VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Systém pro získávání informací o vědeckých publikacích skupin autorů

A System for Retrieving Information to Scientific Publications of Groups of Authors

2017 Bc. Tomáš Kucharczyk

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Tomáš Kucharczyk**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: Systém pro získávání informací o vědeckých publikacích skupin autorů
A System for Retrieving Information to Scientific Publications of Groups of Authors

Jazyk vypracování: čeština

Zásady pro vypracování:

Aktuálně existuje celá řada organizací spravující údaje o vědeckých publikacích ve vlastních informačních systémech resp. databázích publikací (např. Web of Science, Scopus, Google). Tyto informační systémy poskytují často jen omezenou funkcionalitu, typicky neposkytují statistiky publikační činnosti pro skupiny autorů, katedry, případně fakulty. Cílem této práce je návrh a implementace systému umožňujícího snadno a přehledně zobrazit informace, které nejsou v těchto informačních systémech běžně k dispozici.

1. Vyberte alespoň dvě databáze publikací a nastudujte funkcionalitu informačních systémů těchto databází. Nastudujte možnosti identifikace autorů v těchto databázích.
2. Nastudujte aplikační rozhraní těchto databází a vytvořte infrastrukturu pro využití těchto rozhraní.
3. Navrhněte a implementuje systém umožnující získání informací o vědeckých publikací, zaměřte se především na statistiky publikační činnosti pro skupiny autorů, katedry a fakulty. V systému bude možné získávat statistiky pro různé typy publikací, např. konferenční články, časopisecké články apod.
4. Výslednou aplikaci otestujte a vyhodnoťte.

Seznam doporučené odborné literatury:

Podle pokynů vedoucího diplomové práce.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. Ing. Michal Krátký, Ph.D.**

Datum zadání: 01.09.2016

Datum odevzdání: 28.04.2017

doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 24. dubna 2017

Rád bych na tomto místě poděkoval vedoucímu práce **doc. Ing. Michalu Krátkému, Ph.D.** za jeho ochotu při poskytování rad a odborné konzultace při zpracování této diplomové práce. V neposlední řadě bych poděkoval své rodině za podporu a zázemí během celého studia.

Abstrakt

Aktuálně existuje celá řada organizací spravující údaje o vědeckých publikací ve vlastních informačních systémech resp. databázích publikací (např. Web of Science, Scopus, Google). Tyto informační systémy poskytují často jen omezenou funkcionalitu, typicky neposkytují statistiky publikační činnosti pro skupiny autorů, katedry, případně fakulty. Cílem této práce je návrh a implementace systému umožňujícím snadno a přehledně zobrazit informace, které nejsou v těchto informačních systémech běžně k dispozici.

Klíčová slova: informační systémy, vědecké publikace, akademické publikace, databáze publikací, publikační činnost autorů, Scopus, Web of Science

Abstract

Currently, there are a number of organizations managing data of scientific publications on their own information systems with publication databases (such as Web of Science, Scopus, Google). These information systems usually provides limited functionality, typically without publishing statistics for authors, departments or faculties. The aim of this work is to design and implement the system, which can easily and clearly present informations that are not ordinarily available in mentioned information systems.

Key Words: Information systems, Scientific publications, Academic Publications, Databases of publications, Publishing activities of authors, Scopus, Web of Science

Obsah

Seznam použitých zkratek a symbolů

API	–	Access Programing Interface
CRUD	–	Create Read Update Delete
CSS	–	Cascade Styling Sheets
DAO	–	Data Access Object
DB	–	Databáze
DP	–	Diplomová práce
HTML	–	Hyper Text Markup Language
HTTP	–	Hyper Text Transport Protocol
ID	–	Identifikátor
IS	–	Informační systém
JS	–	JavaScript
ORM	–	Object-Relational Mapping
REST	–	Representational State Transfer
SOAP	–	Simple Object Access Protocol
TDG	–	Table Data Gateway
WoS	–	Web of Science
WWW	–	World Wide Web
XML	–	eXtensible Markup Language
XPath	–	XML Path Language

Seznam obrázků

Seznam tabulek

Seznam výpisů zdrojového kódu

1 Úvod

Členové akademických organizací evidují informace o svých vědeckých publikacích v několika organizacích, které ve vlastních informačních systémech udržují různá data a statistiky. Mezi největší databáze takového typu patří například Web of Science, Scopus nebo Google Scholar. Tyto služby nabízí vlastní vyhledávače, ve kterých je možné dohledat informace o autorech, publikacích a jejich citacích. Rozšířená funkcionalita v podobě zobrazování přehledů či statistik pro skupiny autorů (organizace, fakulty, katedry) však ve vyhledávačích těchto systémů chybí. Přehled publikační činnosti pro skupiny autorů VŠB-TUO by však poskytl hodnotné informace.

Cílem této práce je studium možností připojení k databázím publikací, návrh a vytvoření informačního systému, který by dokázal poskytnout požadované funkce pro využití v rámci školy.

Tento projekt je vytvářen v rámci dvou diplomových prací, kdy je tato práce zaměřena na analýzu autorů a jejich publikací. Druhou DP je Systém pro získávání informací o citačním ohlasu publikací pro skupiny autorů [1] od Jiřího Littnera z roku 2017, která se zabývá analýzou citací publikací autorů. Část systému spravující funkce pro citace jsou popsány ve zmíněné DP, proto budou v této práci probrány jen okrajově. Zajištění přístupu ke zdrojovým databázím je rovněž rozděleno. Součástí této DP je přístup k Web of Science, naopak práce Jiřího Littnera se zabývá přístupem k aplikaci Scopus. Znalosti jsou během tvorby sdíleny pro jednotné získání informací o autorech publikací i citací jak pro Web of Science, tak pro Scopus.

DP je rozdělena do šesti kapitol, které směřují k úspěšnému vypracování řešení dle zadání:

1. **Úvod**
2. **Databáze publikací a citací:** úvod do problematiky publikací a jejich databází.
3. **Analýza:** analýza IS, seznam požadovaných funkcí, popis dat a lokální databázi. V závěru nastiňuje obecný popis algoritmů výpočtů statistik.
4. **Návrh:** popis návrhu řešení, architektura, použité technologie a prostředky.
5. **Implementace:** DAO[1] objekty , ORM[2], funkce pro získávání a uchovávání dat, přepočty nad daty a jejich aktualizace.
6. **Formuláře:** popis a ukázka výsledných formulářů.
7. **Testování, měření a zhodnocení aplikace:** popis proběhlých testů, jejich výsledků a statistiky hodnot z proběhlých měření.
8. **Závěr:** závěr a zhodnocení práce.

[1] Data Access Object
[2] Object-Relational Mapping

2 Databáze publikací a citací

2.1 Publikace a citační ohlas

Odborné články vědeckých (resp. akademických) pracovníků se zabývají úzce vymezeným pohledem na určité téma a skýtají mimo vlastního obsahu také řadu metadat. Taková data obsahují informace o jejich autorovi a citacích. Slouží zejména k udržení řádu v nepřeberném množství publikací s různými vlastnostmi a oblastmi výzkumu. Systémy, které tato data uchovávají mají vlastní struktury a pojmenování, avšak účel je stejný. A to poskytnout možnost širokého využití samotných publikací napříč obory a organizacemi.

Citační ohlas (nebo citační index) určitého odborného článku je dán počtem prací v odborných vědeckých časopisech, které na daný článek odkazují. Citační index také slouží ke stanovení dopadu vědeckých časopisů, který udává průměrný počet citací průměrného článku v daném médiu. Citační index může být při splnění jistých podmínek využit jako jedno z pomocných kritérií hodnocení vědecké práce jednotlivce či institucí. Často platí, že proslulí a významní vědci mají články s vyššími citačními indexy. Práce s citačními databázemi tedy patří k nezbytným součástem vědecké práce, zejména pak publikování výsledků výzkumu.

2.2 Databáze vědeckých publikací

Elektronické databáze vědeckých publikací mohou a nemusí být volně přístupné veřejnosti. Vysoké školy, knihovny a další veřejné instituce je však mají předplacené a umožňují tak studentům a pracovníkům vyhledávat odborné články pro akademické účely. Webové aplikace takových systémů obsahují vlastní vyhledávače, ve kterých lze podle různých parametrů vyhledávat. Takovým parametrem může být pro autora jméno nebo identifikátor, v případě publikací název, ISBN, titulek, ID, datum publikování, obor či klíčová slova. Parametry lze zpravidla kombinovat a výběr tak zúžit na konkrétní výsledky.

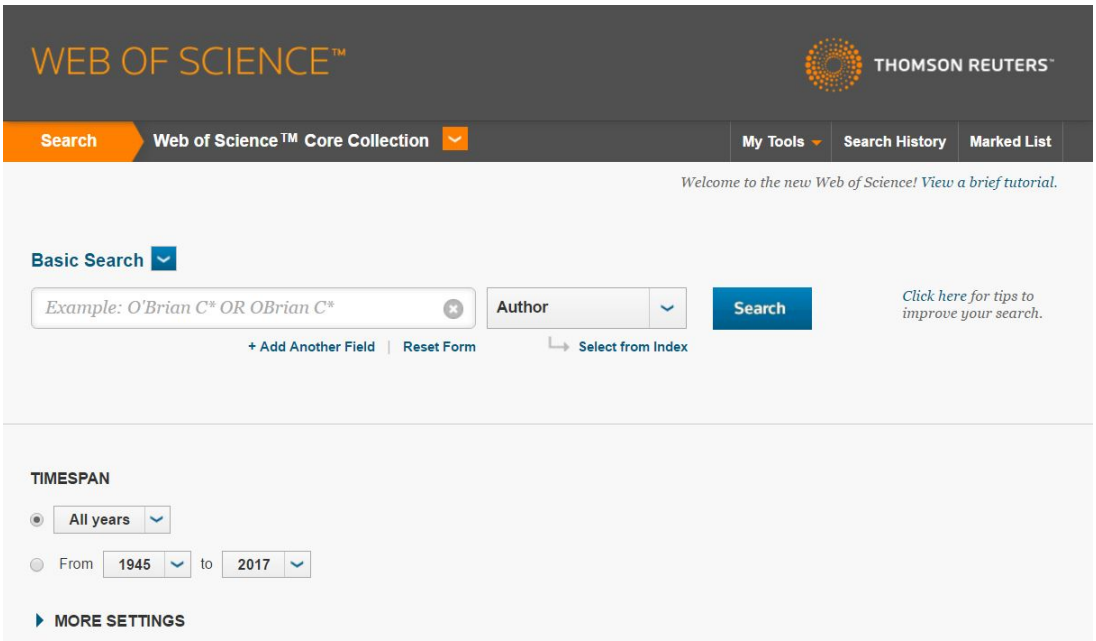
2.2.1 Web of Science Core Collection

Databáze Web of Science Core Collection (dále jen WoS) [2] je významný světový informační zdroj v oblasti výzkumu a vývoje. Online akademická služba provozovaná společností Clarivate Analytics obsahuje údaje o článcích, jejich autorech, obsahu a referencích, citovanosti a edičních údajích. Poskytuje také bibliografické záznamy včetně abstraktů.

WoS poskytuje přístup k 7 databázím:

- Science Citation Index (SCI): přírodní vědy – sleduje citace ve vědeckých časopisech ze 150 oborů přírodních a technických věd od r. 1900, zdrojem je přes 8 500 periodik

- Social Sciences Citation Index (SSCI): společenské vědy - sleduje citace ve vědeckých časopisech z 55 oborů společenských věd.
- Arts Humanities Citation Index (AHCI): umění a humanitní vědy – sleduje citace ve vědeckých časopisech z oborů humanitních věd a uměnovědy, zdrojem je přes 1 700 periodik a dále indexuje vybrané články z více než 250 vědeckých a společenskovědních časopisů – od r. 1975.
- Index Chemicus: přírodní vědy – více než 100 předních světových časopisů o chemii.
- Current Chemical Reactions: Kompletní reakční schémata, kritické podmínky, bibliografické údaje a souhrny pro autory.
- Conference Proceedings Citation Index: Science: přírodní vědy – sleduje citace v konferenční literatuře ze všech oborů přírodních a technických věd.
- Conference Proceedings Citation Index: Social Science and Humanities: společenské vědy a humanitní vědy - sleduje citace v konferenční literatuře ze všech oborů společenských a humanitních věd a uměnovědy.



Obrázek 1: Vyhledávač aplikace Web of Science (Zdroj: http://apps.webofknowledge.com)

Obsah této databáze zahrnuje více než 10 000 časopisů z celého světa včetně Open Access časopisů a více než 110 000 sborníků konferencí. Pokrývá přírodní vědy, sociální vědy, umění a humanitní vědy od roku 1900 ve 256 disciplínách.

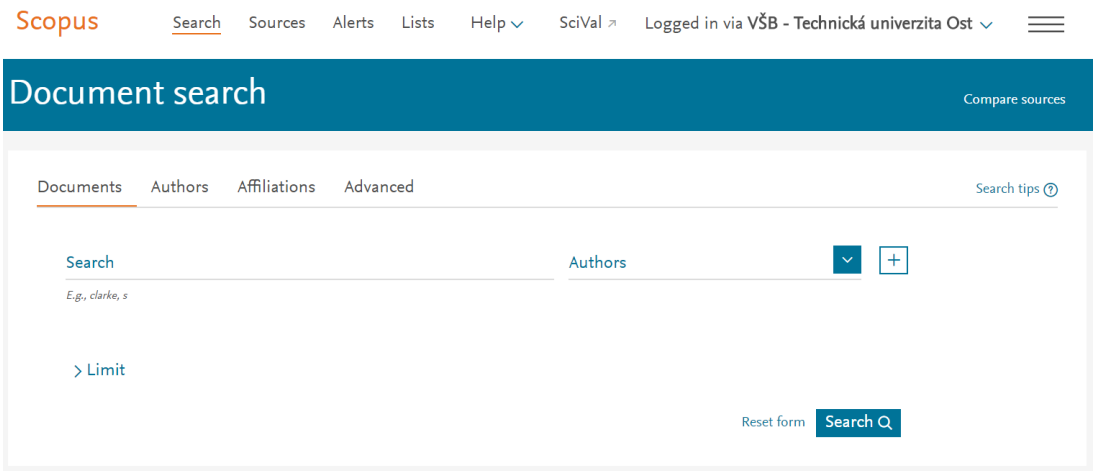
Tato databáze poskytuje přístup k datům pro předplatitele (organizace) a to v podobě vyhledávače webové aplikace nebo API. Komunikace s API probíhá skrze SOAP zprávy.

2.2.2 Scopus

Dalším vybraným zdrojem je databáze Scopus [3], což je databáze publikací a citací odborné literatury, taktéž dostupná pouze pro registrované uživatele, resp. organizace.

Obsahuje abstrakty a záznamy z téměř 20 500 recenzovaných časopisů od více než 5 000 vydavatelů po celém světě. Databáze Scopus obsahuje přímé odkazy na plné texty článků, knihovnické zdroje a další aplikace, jako je například systém pro správu bibliografických referencí.

Podobně jako WoS nabízí i Scopus přístup k datům a to v podobě vlastního webového vyhledávače, či pomocí nabízeného API, které na základě vyžádaných parametrů vrací data v podobě XML.

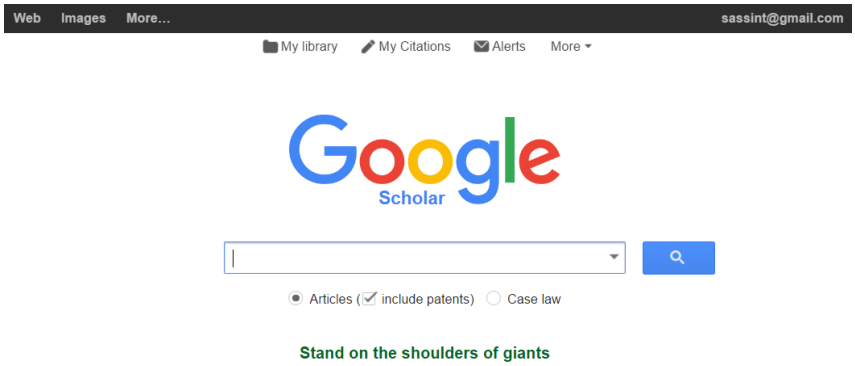


Obrázek 2: Vyhledávač aplikace Scopus (Zdroj: https://www.scopus.com/)

Původně bylo záměrem využít také zdroj Scimago (http://www.scimagojr.com/), avšak analýza ukázala, že tento databázový systém je postaven na systému Scopus. Vzhledem ke stejnému zdroji dat tak není databáze Scimago zařazena mezi další zdroje.

2.2.3 Google Scholar

Google Scholar [4] je volně přístupný webový vyhledávač publikací, který indexuje plná znění nebo metadata akademické literatury různého typu formátů a oborů. Beta verze byla vydána v listopadu 2004. Od té doby jsou postupně přidávány (resp. odebírány) funkcionality. V červnu 2013 byl odebrán z přístupu Google AJAX API, tedy rozhraní pro využití třetími stranami. Google Scholar obsahuje většinu recenzovaných on-line akademických časopisů a knih, konferenčních příspěvků, diplomových a disertačních prací, souhrnů, technických zpráv a další odborné literatury, včetně soudních posudků a patentů.



Obrázek 3: Vyhledávač aplikace Scholar (Zdroj: https://scholar.google.cz/)

2.3 Práce s API

Tato kapitola se věnuje možnostem využití rozhraní databázových systémů, problémům a nástinu řešení pro implementaci nástrojů pro získávání potřebných dat. Nutno dodat, že veškerá komunikace se všemi organizacemi byla profesionální, avšak měla zdlouhavý průběh a pro získání všech informací a vyjádření se vývoj mnohokrát zpomalil, ne-li zcela zastavil. V této kapitole jsou tudíž popsány pouze hlavní části komunikace vedoucí ke konečnému řešení.

2.3.1 Web of Science API

Každý informační systém stojí na datech, proto je prvním krokem jejich obstarání ze zadaných databází. Dostupná dokumentace projektu ISI Web of Knowledge SM Web Services [5] hovoří, jaké funkce API poskytuje. V rámci vyhledávání se WoS API skládá ze tří webových služeb:

- **WokMWSAuthenticate:** Autentizační webová služba pro získání SID

  URL: http://search.isiknowledge.com/esti/wokmws/ws/WOKMWSAuthenticate

- **WokSearchLite:** Základní vyhledávací služba

  URL: http://search.isiknowledge.com/esti/wokmws/ws/WokSearchLite

- **WokSearch:** Rozšířená vyhledávací služba

  URL: http://search.isiknowledge.com/esti/wokmws/ws/WokSearch

Tyto služby mohou s aplikacemi komunikovat skrze protokol SOAP. Kvůli potřebě komunikace mezi IS a databázemi WoS je tedy nutné vytvořit SOAP klienta schopného těchto úkonů. Implementace takového SOAP klienta musí vytvářet SOAP autorizační požadavek pro získání SID nutné pro vyhledávání. Po získání identifikátoru je možné zasílat požadavky na vyhledávací metodu webové služby a takto s databází nadále komunikovat.

Dokumentace API nepopisuje zcela přesně všechny informace, které jsou k implementaci vlastního komunikátoru potřeba a to zejména z předpokladu využití existujících aplikací. Na vyžádání mi však podpora Web of Science poskytla PDF dokument s návodem nastavení komunikátoru SOAP UI, který obsahuje dostatek informací k implementaci vlastního SOAP klienta. Při jeho vývoji je pro simulace a testování možné využít právě řešení SOAP UI, kterým lze ověřit jak tvorbu požadavků, tak shodu výsledků s vlastní implementací klienta.

```
<records>
    <uid>WOS:000356983400002</uid>
   <title> <label>Title</label>
               <value>Cost-based holistic twig joins</value> </title>
   <doctype> <label>Doctype</label>
                <value>Article</value>
                <value>Proceedings Paper</value> </doctype>
   <source> <label>Published.BiblioDate</label>
               <value>AUG-SEP</value> </source>
   <source> <label>Published.BiblioYear</label>
               <value>2015</value> </source>
   <authors> <label>Authors</label>
               <value>Baca, Radim</value>
               <value>Lukas, Petr</value>
               <value>Kratky, Michal</value> </authors>
   <keywords> <label>Keywords</label>
                 <value>XML</value>
                 <value>Query processing</value>
    ...
</records>
```

Výpis 1: Ukázka části XML dokumentu s daty v odpovědi serveru Web of Science

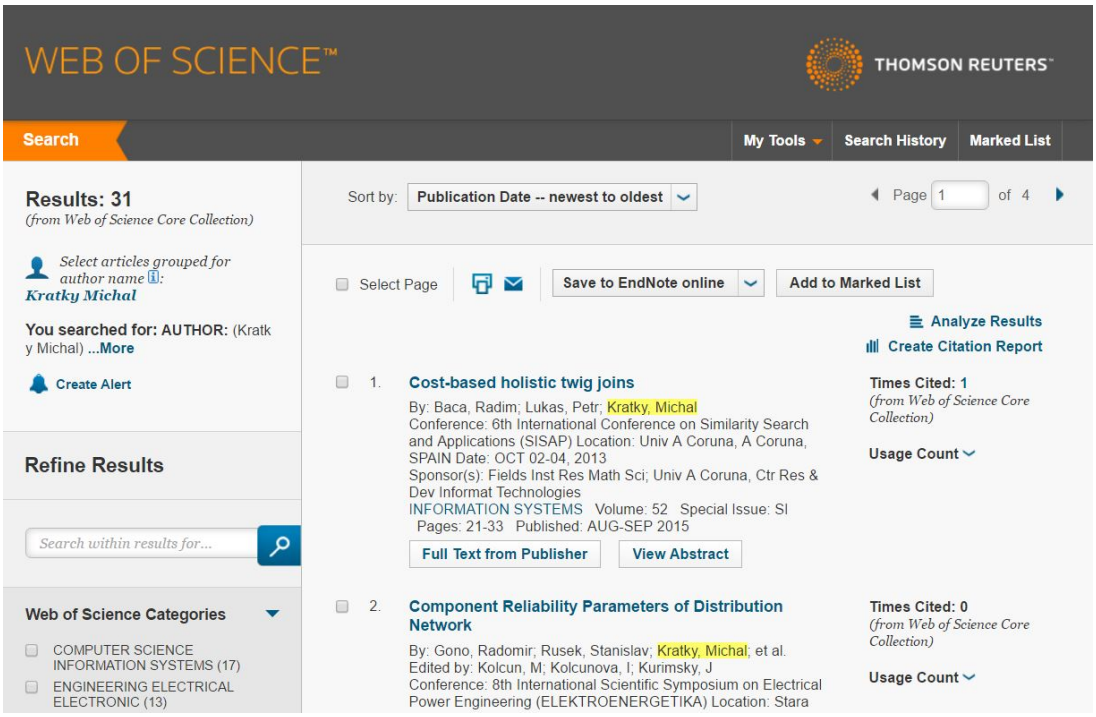
2.3.1.1 Přístup k databázi Během vytváření vlastního SOAP klienta pro přístup k API WoS se ukázalo, že přístup k rozšířené vyhledávací službě WokSearch je uzavřen. Povolená služba WokSearchLite by neposkytla všechny potřebné informace, zejména citační ohlas publikací potřebný pro realizaci DP Jiřího Littnera [1]. Proto jsem začal zjišťovat, jaké kroky jsou potřebné k odemknutí rošířené služby. Z emailové komunikace s podporou produktu Web of Science vyplynulo, že tento požadavek musí být vznesen skrze osobu zodpovědnou za školní knihovnu s povolením osoby zodpovědné za využívání dat (vedoucí diplomové práce).

Následná komunikace s knihovnou a rovněž s Web of Science Account Managerem pro ČR přinesla informace o tom, že VŠB – TUO má předplacen pouze základní přístup bez možnosti

používat WokSearch (rozšířené hledání vč. citací). Rozšířený přístup do WoS Core Collection je ceněn na zhruba 900 tisíc Kč na rok a tento požadavek byl samozřejme knihovnou zamítnut.

Povolený přístup k službe WokSearchLite nicméně není schopný poskytnout veškeré informace potřebné ke splnění požadavků pro vyvíjený informační systém. Těmito informacemi jsou například veškeré informace o citacích nebo detailní informace o publikacích autora (typ dokumentu, ID...). Implementace získávání dat z WoS skrze API je nicméně ponechána v systému pro další užití.

2.3.1.2 Web browser gathering Po zvážení je vybrána možnost poněkud náročná, ale uskutečnitelná a vyhovující, a sice sbírání dat z webového vyhledávače aplikace Web of Science. Tento vyhledávač totiž poskytuje dostatek informací k čerpání do informačního systému. K tomuto účelu je potřeba vytvořit nástroj, který je schopný simulovat průchod webem jako živý uživatel (vyplnění polí,otvírání odkazů, čtení dat).



Obrázek 4: Výsledky vyhledávání (Zdroj: http://apps.webofknowledge.com)

Takový přístup je samozřejmě možný a není v rozporu s licencí o využití dat, kdy jsme omezeni na vyhledávání, využívání a prezentaci dat pouze v rámci organizace VŠB-TUO. Ovšem přináší také další kroky potřebné k celkové realizaci. Je totiž předem zřejmé, že je tento simulovaný průchod webovou aplikací velmi náročný na čas. Jestliže bychom byli SOAP komunikací s API získat např. seznam publikací pro autora s několika detaily, vyřízení požadavku by trvalo řádově vteřiny. Naproti tomu průchod všech prvků stránky automatizovaným nástrojem má sice daleko vyšší rychlost než člověk, ale i tak je doba vyřízení takového požadavku velmi

dlouhá. Pokud má autor v takové databázi stovky záznamů, mohla by se doba vyřízení oproti API pohybovat v hodinách, což je nežádoucí.

Pro získání všech informací potřebných pro uskutečnění zadání jak mé diplomové práce tak práce zabývající se citacemi je nezbytné procházet samotný výhledávač webové aplikace Web of Science a data stahovat. Tento vyhledávač obsahuje základní ovládací prvky jako textové pole pro zadání parametrů či roletku s type vyhledávání podle názvů publikací, klíčových slov, autorů identifikátorů a podobně (viz obrázek 1).

Zadáním vyhledávání je aplikace přesměrována na stránku s výsledky (obrázek 4). Výsledky zobrazují základní informace o jednotlivých publikacích autora. Jednotlivé výsledky odkazují na detailní stránku, na které jsou zobrazeny veškeré informace o publikaci. Velkým urychlením při procházení výsledků je možnost přejít z detailu výsledku rovnou na další položku bez nutnosti přesměrování zpět na seznam všech výsledků. Pro průchod tedy stačí otevřít první výsledek a dále se posouvat tak dlouho, dokud není zobrazena poslední položka. Poté lze pokračovat na domovskou stránku a začít vyhledáváním dalšího autora.

Pro implementaci takového nástroje lze využít třídy HTTPRequest a HTTPResponse [6]. Hotová komponenta WebBrowser [7], která již obsahuje implementaci metod pro zasílání požadavků či navigaci ve webu se rovněž jeví jako vhodná, byť je primárně určena pro kompenentu prohlížeče při použití WinForms [3] aplikací.

2.3.2 Scopus API

Situace na straně databází Scopus je oproti WoS naprosto odlišná. Rozšířené možnosti API [8] jsou sice implicitně rovněž zamčeny, ale pro jejich využití je potřeba pouze požadavek od osoby zodpovědné za školní knihovnu a od osoby zodpovědné za využívání těchto dat, tedy vedoucího práce. Komunikací s podporou Scopus je tedy po čase povoleno plné využití Scopus API v rámci organizace VŠB-TUO a lze tak získat veškerá data jak o publikační činnosti autorů, tak o citačním ohlasu.

K získání informací jsou využity dvě rozhraní: Author Search API a Scopus Search API. Tato API poskytují výsledky ve formě XML, na základě REST API požadavků. Na rozdíl od WoS není potřeba při každém otevírání komunikace požadovat SID, ale ověřování probíhá na základě API Key [4], který je neměnný a je registrován pro organizaci. Tento klíč je přidáván do hlavičky každého vyhledávacího požadavku společně s *querry string* [5].

Rozhraní SCOPUS:

- **Author Search API** - informace o autorovi a organizaci.
- **Scopus Search API** - informace o publikaci (ID, autoři, publikováno, typ dokumentu).

[3] Windows Forms (WinForms) je grafická knihovnam která je součástí .NET Framework.
[4] Identifikační klíč přístupu k rozhraní
[5] Parametry a jejich hodnoty složené do jednoho textového řetězce oddělené speciálním znakem.

První z uvedených API je využívána zejména pro získání identifikátorů autorů a pro další použití při získávání citací. Pro potřeby zmapování publikační činnosti je dostačující Scopus Search API. Z autorizovaného zdroje (API Key) lze zasílat požadavky pro obdržení XML dokumentu s daty. Požadavek pro získání informací (výpis 2) o publikacích autora se skládá z dotazovaných atributů (*field*) a filtrovacího parametru. Filtrovacím parametrem (*querry*) je identifikátor autora (*au-id*). Vytvořený HTTP požadavek doplněný o hodnotu ID autora zaslaný na tuto URL umožní získat odpověď serveru s daty.

```
http:/api.elsevier.com:80/content/search/scopus?field=
dc:identifier,prism:publicationName,prism:coverDate,prism:aggregationType
&query=au-id("")
```

Výpis 2: URL s parametry pro Scopus Search API

Dotazované atributy:

- **dc:identifier** - Identifikátor publikace (Scopus ID).
- **prism:publicationName** - Název publikace.
- **prism:coverDate** - Datum vydání.
- **prism:aggregationType** - Typ dokumentu.

Pro splnění požadavků na tento zdroj je tedy nutné zajistit komunikaci skrze požadavky, nastavení parametrů vyhledávání a také získání informací z odpovedí serveru. Získání výsledků jednoduchých dotazů je zde sice otázkou vteřin, ale při větším objemu dat (tabulka seznamu publikací autora se stovkami záznamů) se doba adekvátně prodlužuje.

Výsledkem jsou objemné XML dokumenty s hodnotami, které je navíc potřeba přečíst a vložit do doménových objektů. Čas pro dokončení takových operací je za hranicí snesitelnosti uživatele informačního systému.

```
<entry>
   <dc:identifier>SCOPUS_ID:84927949718</dc:identifier>
   <dc:title>Cost-based holistic twig joins</dc:title>
   <prism:publicationName>Information Systems</prism:publicationName>
   <prism:issn>03064379</prism:issn>
   <prism:coverDate>2015-08-01</prism:coverDate>
   <prism:aggregationType>Journal</prism:aggregationType>
   <author seq="1">
      <author-url>http://api.elsevier.com/content/author/author_id/6701917792</
          author-url>
      <authid>6701917792</authid>
      <authname>Kratky M.</authname>
```

```
      <surname>Kratky</surname>
      <given-name>Michal</given-name><initials>M.</initials>
   </author>
   ...
</entry>
```

Výpis 3: Ukázka části XML dokumentu s daty v odpovědi serveru Scopus

2.3.3 Google Scholar

Tato zdrojová databáze naneštěstí nenabízí jakékoli API, byť k akademickému použití. Takový přístup Google Scholar nikdy neměl a z vyjádření Google vyplývá, že ani v brzké budoucnosti mít nebude. Nabízí se myšlenka využít stejný princip jako v případě Web of Science, ale v tomto případě nelze stejný postup uplatnit kvůli podmínkám použití aplikace. Tento zdroj je v rámci diskuse s vedoucím práce zamítnut a nebude součástí řešení.

2.3.4 Identifikátory a podpora API funkcí

Při prohledávání zdrojových databází za účelem získávání informací a následném vytvoření přesných statistik je nutné zaměřit se na přesnou identifikaci. Vyhledávání publikací autorů v těchto systémech probíhá pomocí jména a příjmení autora nebo přiděleného identifikátoru. Jména se v databázi opakují, proto je pro vyhledávání mnohem vhodnější použít identifikátory.

Identifikátory autorů:

- **Researcher ID**
  - vlastní identifikační systém Web of Science pro autory - http://www.researcherid.com/.
- **OrcID** (Open Researcher and Contributor ID)
  - používáno ve Scopus pro autory - https://orcid.org/.

Pro identifikaci jednotlivých publikací je možné využít hodnoty přiřazené každému záznamu.

Identifikátory publikací:

- **ScopusID**: označení publikace Scopus.
- **WOSID**: označení publikace WoS.
- **ISBN**[6]: standard pro unikátní 13místný číselný identifikátor publikací
- **ISSN**[7]: osmimístný číselný identifikátor pro jednotlivé díly knižní série

[6] International Standard Book Number
[7] International Standard Serial Number

Následující tabulka zobrazuje přehled podpory API databází a také možnosti průchodu webem WoS. Lze z ní vyčíst, že Web of Science API [5] neposkytuje vyhledávání publikací pro autora dle identifikátoru (nepřesnost) a rovněž neposkytuje informace pro citační ohlas publikací. Ostatní dva zdroje tato data poskytují.

Vyhledávání	WoS API	WoS Browsing	Scopus API
Autoři			
ID autora	ne	ano	ano
Jméno	ano	ano	ano
Publikace			
Název	ano	ano	ano
ID publikace	ano	ano	ano
Publikováno	ano	ano	ano
Typ dokumentu	ano	ano	ano
Citace			
Citace publikací	ne	ano	ano
Citace bez vlastních	ne	ano	ano

Tabulka 1: Tabulka podporovaných funkcí datových zdrojů

3 Analýza

3.1 Požadavky na informační systém

Vytvořením systému pro získávání informací o vědeckých publikacích skupin autorů lze uživatelům z řad VŠB-TUO poskytnout souhrnné informace o autorech vědeckých publikací. Jednoduchá webová aplikace musí obsahovat prezentační část s agregovanými daty a administrativní část pro správu IS.

3.1.1 Seznam funkcí

1. **Seznam informací o autorech:** ResearcherID (WoS), ORCID (Scopus), fakulta, katedra, skupina, typ pracovníka.
2. **Souhrnná publikační činnost autora:** Identifikátory autora, celkový počet článků v časopisech a ve sbornících konferencí, citační ohlas publikací.
3. **Celková publikační činnost autora:** Celkový počet článků v časopisech, ve sbornících konferencí, jednotlivé počty za posledních 5 let, součet za posledních 5 let a součet z let ostatních. Možnost zobrazení pro jednotlivé nebo všechny zdroje.
4. **Celkový přehled publikační činnosti kateder:** Viz bod 3 pro jednotlivé katedry.
5. **Úprava a rozšiřitelnost seznamu autorů a struktury organizace** Přidání a úprava stávajících autorů, kateder a skupin.
6. **Stahování a agregace dat v pravidelném časovém intervalu** [8] Pro udržení aktuálnosti dat je třeba zajistit pravidelné aktualizace a to tak, aby nikdy nebyla uživateli zobrazena v nekonzistentním stavu.

3.1.2 Uživatelské role

Dle zadání vyplývají pro práci se systémem dvě uživatelské role: návštěvník a administrátor.

- **Návštěvník**
  - Běžný uživatel aplikace.
  - Práva ke čtení.
  - Má přístup k přehledům a analýzám.
- **Administrátor**
  - Správce aplikace.

[8] Požadavek doplněn během vývoje.

- Práva pro čtení a aktualizace.
- Má přístup k přehledům, analýzám a administrativní části.
- Spravuje strukturu organizace a seznam autorů.

3.1.3 Zobrazení dat

Data jsou zobrazována skrze tři formuláře: souhrnný přehled, autoři a citace. Souhrnným přehledem je míněna tabulka autorů s uvedenými hodnotami pro publikování v různých databázích pro daná období. Takový náhled slouží k základnímu získání informací a srovnání autorů. Zbývající formuláře zastupují detailní výpisy informací autorů pro každou ze zdrojových databází (WoS,Scopus). Veškerá prezentovaná data je možné řadit dle různých parametrů, ale také třídit - např. podle hierarchie organizace.

Administrativní část obsahuje funkce systému, která pracují s daty IS. Lze zde přidávat, odebírat a jinak upravovat jak autory, tak strukturu organizace (fakulty, katedry). Rovněž jsou zde obsaženy funkce systému, které se starají o aktualizace a operace nad daty.

Systém musí být dostatečně rychlý a použití přehledné a intuitivní.

3.2 Data a lokální databáze

Původním záměrem práce bylo zajistit zobrazování výsledků z API v reálném čase, ale vzhledem k nutnosti použít průchod webem Web of Science je doba vyřízení takového požadavku velmi dlouhá. Aby se takto náročný proces nemusel využívat při každém dotazu, je nutné vytvořit lokální databázi systému 3.2. Do této databáze jsou stahována data, která obsahují základní informace o publikační činnosti autorů a jejich citačním ohlasu. Takto uložená data lze využít pro vlastní agregace a následnou prezentaci. Použitím takové databáze se několikanásobně zkrátí doba zobrazení dat uživateli IS.

Pro účely ukládání získaných dat je v aplikaci vytvořena databáze, která uchovává informace o autorech, publikacích a jejich citacích. Autoři jsou do lokální databáze naimportováni ze seznamu poskytnutého vedoucím práce, kdy tento seznam po doplnění obsahuje jméno, login, identifikátor Scopus (OrcID), identifikátor Web of Science (ResearcherID), zařazení pod fakultu, katedru či odbornou skupinu. Funkce administrativní části sytému poskytují operace nad tabulkou autorů, kde je potřeba doplnit autorům ORCID a ResearcherID pro správnou funkci auktualizací dat a pro zamezení duplikací v databázi.

Požadována je rovněž možnost dodávání dalších položek jak autorů a jejich organizačního zařazení, tak struktury organizace v podobě fakult a kateder. Aktualizační funkce si z těchto informací jsou schopny získat a uložit data do databáze.

Po aktualizaci dat je nutné spustit provedení kalkulace agregačních funkcí, které obstarávají data pro přehledy. Všechny tyto funkce jsou popsány v kapitole Obecný popis algoritmů výpočtů 3.3.

K vytvoření analytických přehledů pro autory a jejich publikace je potřeba získat informace o všech publikacích pro každého z autorů a to bez vzniku duplicitních záznamů v databázi. Je totiž běžné, že má publikace více autorů, což je potřeba brát v úvahu při sumarizaci počtů pro skupiny. Pokud tedy existuje například publikace se třemi autory z organizace VŠB-TUO (tedy v seznamu autorů IS), není možné pro každého z nich započítat publikaci v rámci skupiny, aby nebyl dokument započítán třikrát, ale pouze jednou.

Následuje popis dat potřebných pro výpočty.

3.2.1 Publikační činnost autorů

Pro bezchybnou analýzu publikací autorů z různých zdrojů je nutné vyhledávat podle identifikátoru autora, nikoliv podle jména. Prvotní skupinou v systému jsou členové Fakulty elektrotechniky a informatiky. Tento seznam autorů poskytnutý vedoucím práce obsahuje 197 autorů a jsou v něm obsaženy základní informace:

- Jméno a příjmení.
- LDAP login [9].
- Identifikátor autora pro Scopus.
- Identifikátor autora pro Web of Science.
- Zařazení pod fakultu.
- Zařazení pod katedru.

Je zřejmé, že zdroj Web of Science nebude používat identifikátory Scopus, proto je nutné pro daný seznam lidí obstarat jejich identifikátory i pro tuto databázi, což by mělo být umožněno administrátorovi systému.

S takto kompletním seznamem informací pro autory lze bezchybně vyhledávat publikace a analyzovat jejich informace. Takovými informacemi jsou zejména:

- Název publikace.
- Rok publikování.
- Typ dokumentu (sborníky konferencí, časopisy).
- ID publikace ve zdrojové databázi.

Získaný seznam publikací autora lze dále použít pro agregační výpočty. Sumarizace publikační činnosti pro autora se dělí podle zdroje dat a podle typu dokumentu, aby bylo možné

[9] Login používaný v systémech VŠB - TU Ostrava.

sledovat počet publikací ve sbornících konferencí nebo časopisech. Pro každou zdrojovou databázi tak budou odděleně zobrazeny dva počty součtu publikovaných dokumentů pro každého autora.

Dále je požadována kategorizace v rámci data publikování se zaměřením na rok. Zobrazené hodnoty jsou následující: součet publikací za jednotlivé roky za posledních 5 let; součet publikací za posledních 5 let; součet ostatních publikací (starších 5 let).

Výsledná analýza autora bude tedy obsahovat:

- Pořadí v rámci aktuálního seřazení.
- Jméno a příjmení autora.
- Identifikátory pro zdrojové DB[10].
- Počty pro zdrojové DB, rozděleno dle typu dokumentu.
- Počty publikací pro jednotlivé roky za posledních 5 let.
- Součet publikací za posledních 5 let.
- Součet ostatních publikací.

Přehled pro souhrnnou publikační činnost autorů s méně detailním pohledem zobrazuje pro jednotlivé autory pouze celkové součty pro zdrojové databáze a je doplněn o citační ohlas publikací autora.

3.2.2 Souhrn publikací skupin autorů

Vzhledem k možnosti zobrazení analýzy pro seznam autorů napříč skupinami (např. pro celou fakultu) je vhodné vytvořit rovněž přehled nad tímto seznamem a to jako součty pro katedry či odborné skupiny. Tento přehled obsahuje stejné součtové sloupce jako přehled pro autory. Namísto jména a identifikátoru je zobrazen název skupiny, příp její označení (kód katedry).

Přehled publikační činnosti skupin je zobrazen nad přehledem autorů po zobrazení výsledků vyhledávání.

Analýza publikační činnosti skupin obsahuje:

- Pořadí v rámci aktuálního seřazení.
- Název skupiny.

[10] publikační databáze WoS, Scopus...

- Označení skupiny.
- Počty pro zdrojové DB, rozděleno dle typu dokumentu.
- Počty publikací pro jednotlivé roky za posledních 5 let.
- Součet publikací za posledních 5 let.
- Součet ostatních publikací.

3.2.3 Duplikace publikací a konzistence dat

Pro zamezení získávání duplicitních záznamů publikací pro skupinu autorů je nutné při analýze každé publikace zjistit, zda už nebyla analyzována a zda se má započítat do součtu počtu publikací. Jelikož jsou součástí analýzy i jedinečné identifikátory publikací každé ze zdrojových databází, je pokaždé potřeba provést kontrolu jedinečnosti. Pokud už byla jednou publikace započítána, nebude dále analyzována a zařazena do celkových počtů pro skupinu, ale pouze pro autora.

Dalším možným problémem může být nekonzistence dat. S tímto problémem může nastat situace, kdy například souhrný počet publikací autora zobrazuje méně publikací než lze vidět v jejím seznamu autorů. Tomu lze zabránit pouze opětovným přepočítáním všech dat (ta nesmí obsahovat redundance) po každé změně.

3.2.4 Aktualizace dat a dočasné úložiště

Pravidelné aktualizace dat je potřeba provést tak, aby se výsledky při stahování dat neukládaly do primárních tabulek a nenarušovaly tak agregační výpočty. Databáze tedy musí obsahovat i záložní tabulky potřebné při aktualizacích, jejichž obsah je po úspěšném ukončení aktualizací a přepočtů překopírován do primárních tabulek. Operace kopírování nepřesáhne dobu několika vteřin.

Časový interval samotné aktualizace je možné nastavit v administrativní části IS. Systémová data tak budou udržována aktuální.

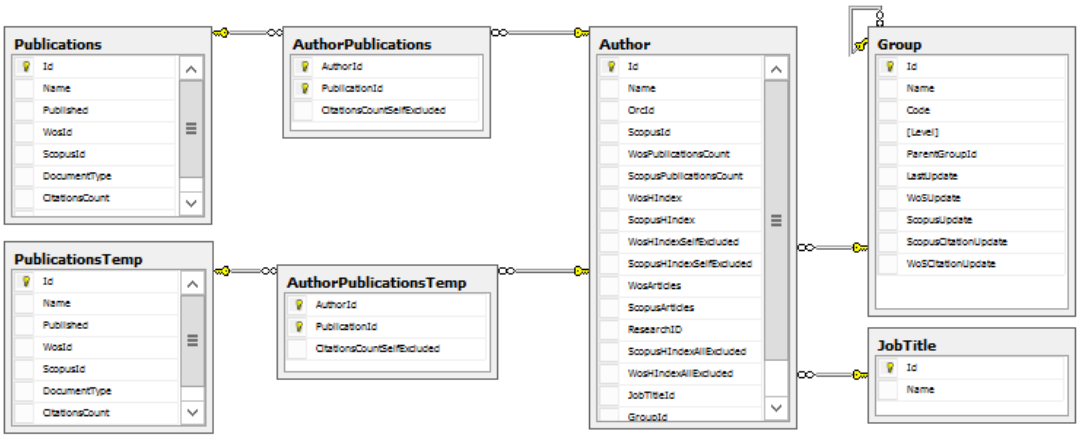
Je nutné vytvořit nástroj, který bude v těchto intervalech spouštět aktualizaci dat a v případě výpadku opakovat celou operaci. Data jsou během aktualizace ukládána do dočasných tabulek a po úspěšném ukončení překopírována do primárních tabulek.

3.2.5 Modelování databáze

Databáze pro uchovávání informací o autorech a publikacích se skládá ze sedmi tabulek (obrázek 11). Hlavní tabulky nesoucí stahovaná data jsou *Author* a *Publication*, které jsou doplněny o vazební tabulky. *Group* a *JobTitle* slouží k ukládání informací o struktuře organizace. Tabulky *PublicationTemp* a *AuthorPublicationTemp* jsou využívány jako dočasné úložiště při aktualizaci dat. Veškeré tabulky se vztahem M:N jsou rozšířeny o spojovací tabulky.

Atributy tabulky dbo.Author:

- **Id** - Identifikátor záznamu.
- **Name** - Jméno autora.
- **OrcId** - Identifikátor ORCID.
- **ScopusId** - Identifikátor Scopus.
- **WosPublicationsCount** - Počet publikací WoS.
- **ScopusPublicationsCount** - Počet publikací (sborníky) Scopus.
- **WosHIndex** - HIndex autora pro WoS.
- **ScopusHIndex** - HIndex autora pro Scopus.
- **WosHIndexSelfExcluded** - HIndex autora pro WoS bez vlastních citací.
- **ScopusHIndexSelfExcluded** - HIndex autora pro Scopus bez vlastních citací.
- **WosArticles** - Počet publikací autora (časopisy) ve WoS.
- **ScopusArticles** - Počet publikací autora (časopisy) ve Scopus.
- **ResearchID** - Identifikátor WoS.



Obrázek 5: Diagram lokální databáze

Atributy tabulky dbo.Publication:

- **Id** - Identifikátor záznamu.
- **Name** - Název publikace.

- **Published** - Rok vydání.
- **WosId** - WoS identifikátor publikace.
- **ScopusId** - Scopus identifikátor publikace.
- **DocumentType** - Typ dokumentu.
- **CitationsCount** - Počet citací publikace.
- **CitationsCountAllAuthorsExcluded** - Počet citací publikace bez autorských.

První data, která jsou do databáze uložena, jsou autoři Fakulty elektrotechniky a informatiky. Fakulta je rozdělena do 7 kateder se 197 autory. Seznam těchto autorů je poskytnut vedoucím práce a je naimportován tak, aby byla zachována hierarchie fakulty.

3.3 Obecný popis algoritmů výpočtů

Následuje obecný slovní popis algoritmů potřebných funkcí v rámci práce s daty (autoři, publikace, citace).

1. Specifikace volby skupiny autorů.
    - Pro vybranou skupinu a zařazení zobraz autory.
2. Sběr dat o publikacích autorů: pro sběr dat IS je možné v cyklech projít strukturu autorů a publikací.
    - Pro každého autora vyhledej publikaci dle ID.
        - Pro každou publikaci z výsledku získej informace a počítej typ publikace.
        - Ulož publikaci s počty do DB.
    - Přepočítej publikace pro autora.
    - Ulož autora s počty do DB.
3. Přepočet autorů: Publikace jsou pro každého autora přepočítány dle typu dokumentu a let vydání.
    - Pro každou publikaci autora.
        - **Je typu sborník konferencí?** Inkrementuj počítadlo sborníků.
        - **Je typu časopisecký článek?** Inkrementuj počítadlo časopisů.
        - **Dle roku publikování zařaď:**
            * starší 5 let,

        * mladší 5 let,
        * kategorizace posledních 5 let (např. 2017/2016/2015/2014/2013).
    - Přepočítej publikace pro autora.
    - Ulož autora s počty do DB.

4. Přepočet kateder: publikace jsou pro každou katedru přepočítány dle typu dokumentu a let vydání.
    - Pro každou skupinu autorů katedry
        - **Je typu sborník konferencí?** Inkrementuj počítadlo sborníků.
        - **Je typu časopisecký článek?** Inkrementuj počítadlo časopisů.
        - **Dle roku publikování zařaď:**
            * starší 5 let,
            * mladší 5 let,
            * kategorizace posledních 5 let.

5. Odstranění duplicit Po uložení dat po stažení se v databázi objevují některé publikace vícekrát, když mají společné autory ze skupiny. Tato funkce duplicity odstraní.
    - Nalezení všech publikací se stejným identifikátorem zdrojové databáze (Scopus, WoS).
    - Vybrání prvního identifikátoru záznamu v rámci lokální databáze (Publication ID).
    - Nahrazení ID pro autorské publikace vybraným ID (přesměrování ze všech duplicitních na jedinou).
    - Smazání duplicitních publikací mimo vybranou.

6. Aktualizace dat v tomto pořadí (v závorce parametry).
    (a) Publikace:
        - aktualizace publikací WoS (autoři),
        - aktualizace publikací Scopus (autoři).
    (b) Citace:
        - aktualizace citací WoS (autoři),
        - aktualizace citací Scopus (autoři).
    (c) Odstranění duplicit.
    (d) Přepočty publikací a citací:
        - přepočty dat WoS (autoři, publikace),
        - přepočty dat Scopus (autoři, publikace).
    (e) Aktualizace tabulek.

4 Návrh

V této kapitole jsou popsán návrh a popis technologií využitých k vytvoření informačního systému popsanému v zadání.

4.1 Použité technologie a prostředky

Pro vytvoření informačního systému je potřeba využít řadu technologií a nástrojů. Ty jsou nyní popsány v jednotlivých podkapitolách a jsou rozebrány z hlediska vlastností, použitelnosti a zejména pro účelné využití k dosažení výsledků.

V rámci výběru technologií pro implementaci projektu byla brána v úvahu zkušenost s vývojem, využitelnost a v neposlední řadě licenční dostupnost takových řešení. Tyto faktory splňuje .NET Framework s využitím MSDN AA [9] licence pro školní použití. Produkty Microsoft jsou proto použity jak pro databázi (MS SQL), tak pro doménovou logiku a prezentaci dat (C Sharp, ASP.NET WebForms).

4.1.1 Aplikační server

Server, na kterém je aplikace nasazena a který je odpovědný za vyřizování požadavků HTTP od uživatelů se nachází na dbedu.cs.vsb.cz v rámci studentského účtu. Školou uvolněné prostředky jsou dostačující pro běh aplikace a to na webovém serveru Microsoft IIS. Server pracuje s OS Microsoft Server 2012 R2.

Obsluha a nasazování na server jsou umožněny skrze VPN [11] připojení do školní sítě s použitím publikačních funkcí Visual Studio [10].

4.1.2 Vývojové prostředí a správa zdrojových kódů

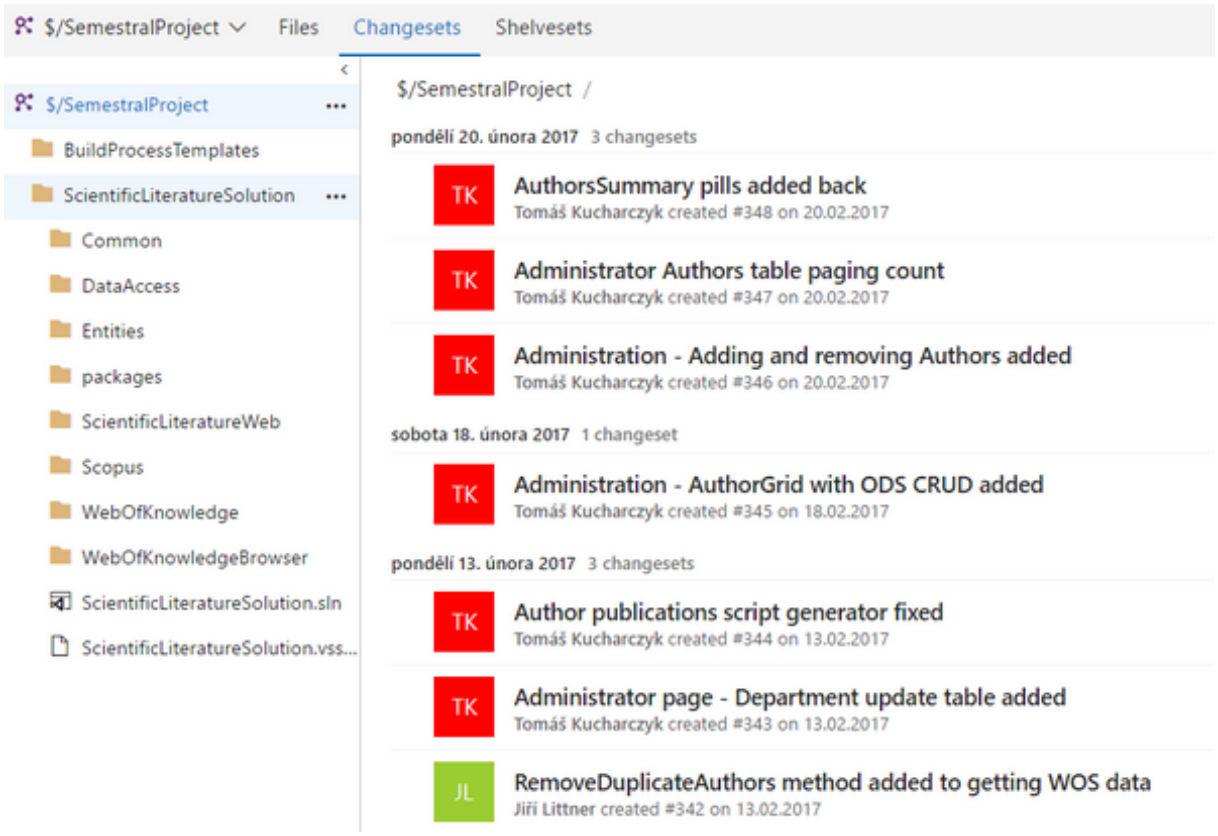
Vybraná platforma .NET nabízí nepřeberné množství možností pro tvorbu dynamických webových stránek skrze IDE[12] Visual Studio. Pro účely DP je použito Visual Studio 2015 ze čkolní licence MSDN. Použitím VS lze využít různé typy ASP.NET aplikací jako Webforms, MVC či WebAPI nabízí funkcionality pro různé typy projektů.

Jelikož je součástí DP práce ve dvou lidech na jedné implementaci projektu, je nutné vyřešit problém správy zdrojových kódů. Tomuto účelu vyhovuje MS Team Foundation Server s mnoha funkcemi, které pokrývají celý životní cyklus software.

URL projektu na TFS: https://tomasjirka.visualstudio.com/SemestralProject

TFS poskytuje: správu zdrojových kódů (společně s Team Foundation Version Control nebo Git), reporting, správu požadavků, projektové řízení (jak pro agilní metodiky tak pro vodopádové), automatické sestavování, vedení vývoje, funkce pro testování či správu vydávání aplikací.

[11] Virtual Private Network - privátní virtuální síť v rámci internetu
[12] Integrated Development Environment - vývojové prostředí



Obrázek 6: Aplikace TFS - ukázka seznamu změn v projektu

Pokrývá celý životní cyklus aplikací a umožňuje použití DevOps. TFS může být použit jako back-end k mnoha integrovaným vývojovým prostředím (IDE), ale primárně je přizpůsoben pro Microsoft Visual Studio a Eclipse na všech platformách.

Tuto službu jsem poprvé poznal a začal využívat při práci pro softwarovou firmu a zanedlouho jsem byl přesvědčen o jejích přednostech a kvalitách. Vývoj ve vícečlenném týmu s TFS (nebo podobných) je velice účinným způsobem práce. Systém check-in and check-out s verzováním historie a snadná možnost začlěnění agilních metod (např. SCRUM [13]) do vývoje je velice učinným způsobem, jak efektivně a přehledně vyvíjet software i v početnějších týmech.

4.1.3 Ostatní technologie a nástroje

- **XML**

  Extensible Markup Language (zkráceně XML, česky rozšiřitelný značkovací jazyk) [11] je obecný značkovací jazyk, který byl vyvinut a standardizován konsorciem W3C. Je zjednodušenou podobou staršího jazyka SGML. Umožňuje snadné vytváření konkrétních značkovacích jazyků (tzv. aplikací) pro různé účely a různé typy dat. Používá se pro serializaci

[13] Iterativní agilní metoda vývoje.

dat, v čemž soupeří např. s JSON či YAML. Zpracování XML je podporováno řadou nástrojů a programovacích jazyků.

Jazyk je určen především pro výměnu dat mezi aplikacemi a pro publikování dokumentů, u kterých popisuje strukturu z hlediska věcného obsahu jednotlivých částí, nezabývá se vzhledem. Prezentace dokumentu (vzhled) může být definována pomocí kaskádových stylů. Další možností zpracování je transformace do jiného typu dokumentu, nebo do jiné aplikace XML.

Jazyk XML na jednu stranu nevyužívá příliš efektivně využitou paměť, ale na druhou stranu je velmi jednoduchý, přehledný a pro člověka čitelný bez nutnosti jakékoli transformace.

Stránky technologie: https://www.w3.org/XML/

- **HttpRequest a HttpResponse**

  Internetové aplikace lze rozdělit zhruba do dvou druhů: klientské aplikace, které požadují informace a serverové aplikace, které reagují na žádosti o informace od klientů. Klasickou internetovou službou klient-server je WWW, kde uživatelé používají prohlížeče pro přístup k dokumentům uloženým na webových serverech po celém světě. Základem komunikace je zaslání požadavku (*request*) ze strany klienta a získání odpovědi (*response*) od serveru skrze protokol HTTP.

  Tímto způsobem lze zajistit komunikaci jak s webovou aplikací tak s API zdrojových databází.

  Stránky technologie: https://msdn.microsoft.com/en-us/library/system.web.httprequest (v=vs.110).aspx

- **Bootstrap**

  Bootstrap je jednoduchá a volně stažitelná sada nástrojů pro tvorbu webu a webových aplikací. Obsahuje návrhářské šablony založené na HTML a CSS, sloužící pro úpravu typografie, formulářů, tlačítek, navigace a dalších komponent rozhraní, stejně jako další volitelná rozšíření JavaScriptu. Pro použití Bootstrapu jsou nutné základní znalosti HTML a CSS, interaktivní prvky jako jsou tlačítka, boxy, menu a další kompletně nastavené a graficky zpracované elementy je totiž možné vložit pouze pomocí HTML a CSS.

  Tato knihovna je v projektu využita pro prezentační vrstvu zejména díky své jednoduchosti, příjemnému vzhledu a rovněž úspoře času.

  Stránky technologie: http://getbootstrap.com/

- **ASP.NET**

  ASP.NET [12] je součást .NET Frameworku pro tvorbu webových aplikací a služeb. Je založen na CLR (Common Language Runtime), který je sdílen všemi aplikacemi postavenými na .NET Frameworku. Projekty tak lze implementovat v jakémkoliv jazyce podporujícím CLR, např. Visual Basic .NET, JScript.NET, C Sharp, Managed C++, ale i mutace Perlu, Pythonu a další. Aplikace založené na ASP.NET jsou také rychlejší, neboť jsou předkompilovány do jednoho či několika málo DLL souborů, na rozdíl od ryze skriptovacích jazyků, kde jsou stránky při každém přístupu znovu a znovu parsovány.

  Koncept ASP.NET WebForms ulehčuje programátorům přechod od programování klasických aplikací pro Windows do prostředí webu: stránky jsou poskládány z objektů, ovládacích prvků (Controls), které jsou protějškem ovládacích prvků ve Windows. Při tvorbě webových stránek je tedy možné používat předpřipravené ovládací prvky jako tlačítka, tabulky a další. Těmto komponentám lze přiřazovat určité vlastnosti, zachytávat na nich události atd. Tak, jako se ovládací prvky pro Windows vykreslí do formulářů na obrazovku, webové ovládací prvky produkují HTML kód, který tvoří část výsledné stránky poslané do klientova prohlížeče.

  Tato technologie je vybrána pro prezentaci dat kvůli prvotnímu rozhodnutí vytvořit projekt v.NET Frameworku a pro účelné využití technologií ASP.NET WebForms.

  Stránky technologie: https://www.asp.net/

- **C#**

  C# [13] je vysokoúrovňový objektově orientovaný programovací jazyk vyvinutý firmou Microsoft zároveň s platformou .NET Framework, později schválený standardizačními komisemi ECMA (ECMA-334) a ISO (ISO/IEC 23270). Microsoft založil C Sharp na jazycích C++ a Java (a je tedy nepřímým potomkem jazyka C, ze kterého čerpá syntaxi). C Sharp lze využít k tvorbě databázových programů, webových aplikací a stránek, webových služeb, formulářových aplikací ve Windows, softwaru pro mobilní zařízení (PDA a mobilní telefony) a dalších.

  Tento jazyk je velmi rozšířený a zároveň je mým primárním programovacím jazykem, proto je vybrán pro implementaci bussiness logiky webové aplikace, včetně komunikace s MS SQL databází.

  Stránky technologie: https://msdn.microsoft.com/en-us/library/z1zx9t92.aspx

- **MS SQL Management Studio**

  SQL Server Management Studio [14] poprvé vydano v roce 2005 je nástroj určený k tvorbě, konfiguraci a správě komponent Microsoft SQL Server. Tento nástroj poskytuje skriptovací

editor a nástroje, které pracují s objekty a funkcemi serveru. Centrální funkcí nástroje SSMS je Object Explorer, který uživateli umožňuje procházet, vybírat a různě manipulovat s objekty na serveru. SSMS Express edition je volně ke stažení a lze se tak připojit a spravovat instanci SQL Server Express.

Od verze 11 používá aplikace WPF technologii stejně jako Visual Studio od verze 2010. Od června 5015 je SSMS vydáváno nezávisle na balíčku SQL Server database engine.

Stránky technologie: https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms#sql-server-management-studio

- **SOAP**

  SOAP [15] (původně Simple Object Access Protocol) je protokolem pro výměnu zpráv založených na XML přes síť, hlavně pomocí HTTP. Formát SOAP tvoří základní vrstvu komunikace mezi webovými službami a poskytuje prostředí pro tvorbu složitější komunikace. Existuje několik různých druhů šablon pro komunikaci na protokolu SOAP. Nejznámější z nich je RPC šablona, kde jeden z účastníků komunikace je klient a na druhé straně je server. Server ihned odpovídá na požadavky klienta.

  SOAP je nástupce XML-RPC, ačkoliv si zapůjčuje jeho způsob přenosu dat a další vlastnosti. Obálka, hlavička a tělo komunikace je ale pravděpodobně z WDDX. Původně ho navrhl Dave Winer, Don Box, Bob Atkinson a Mohsen Al-Ghosein v roce 1998 za podpory firmy Microsoft (kde tou dobou Atkinson a Al-Ghosein pracovali). Dnes je SOAP specifikace držena XML skupinou tvořící internetové protokoly z W3C konsorcia.

  Komunikace s API Web of Science je založena právě na tomto protokolu.

  Stránky technologie: https://www.w3.org/TR/soap/

- **SOAP UI**

  SOAP UI [16] je open-source testovací aplikace pro webové služby postavené na SOA (Service Oriented Architecture) a REST (Representational State Transfer). Byla vydána v roce 2005 na SourceForge a její funkce pokrývají inspekci webových služeb, vývoj, volání, simulace a mocking (tvorba "falešných"objektů v rámci testování části kódu), funkční testování nebo testování vytížení. Využívání je zdarma, je lincencována pod EUPL (European Union Public License) a od vydání byly zaznamenány více než 2 miliony stáhnutí. Existuje také placená verze programu SOAP UI Pro, která se zaměřuje zejména na funkce navrhnuté pro zvýšení produktivity.

  V tomto projektu je SOAP UI využíván pro testování a simulace komunikace s Web of Science databází skrze SOAP komunikační protokol.

  Stránky technologie: https://www.soapui.org/

- **Chrome**

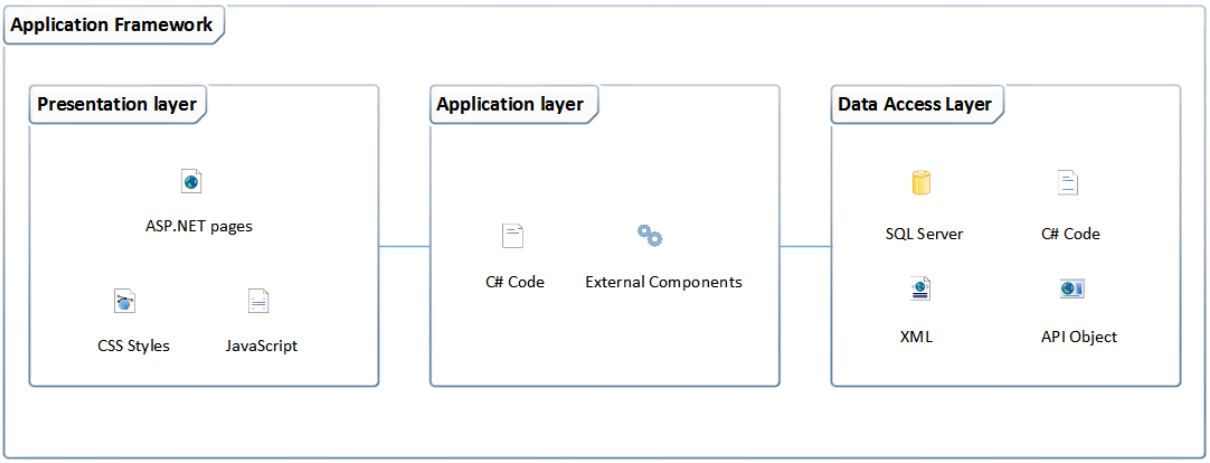
  Rozšířený webový prohlížeč společnosti Google vydaný v roce 2008 je mezi uživateli oblíbený pro své možnosti rozšíření a použití na více platformách. V této práci je prohlížeč využíván pro testování a vývoj informačního systému.

  Dalším s přínosů Google Chrome [17] jsou možnosti pro vývojáře, které přináší spoustu možností při práci s internetovými technologiemi. V tomto nástroji lze ladit webové aplikace, sledovat síťový provoz, odchytávat požadavky a odpovědi serveru či zkoumat funkcionalitu webu.

  Stránky technologie: https://www.google.com/chrome/browser/desktop/index.html

4.2 Architektura

IS je postaven na třívrstvé architektuře a využíva obecné principy pro budoucí modifikace a rozšíření. Jedním z požadavků je, aby projekt obsahoval knihovnu API Object, který lze využít i v jiných implementacích bez nutnosti úprav. Tento objekt musí mít jednoduchou možnost rozšíření o další publikační databáze, jejichž data se přepočítávají a ukládají do lokalní databáze nebo se zobrazují uživateli.



Obrázek 7: Vrstvy informačního systému

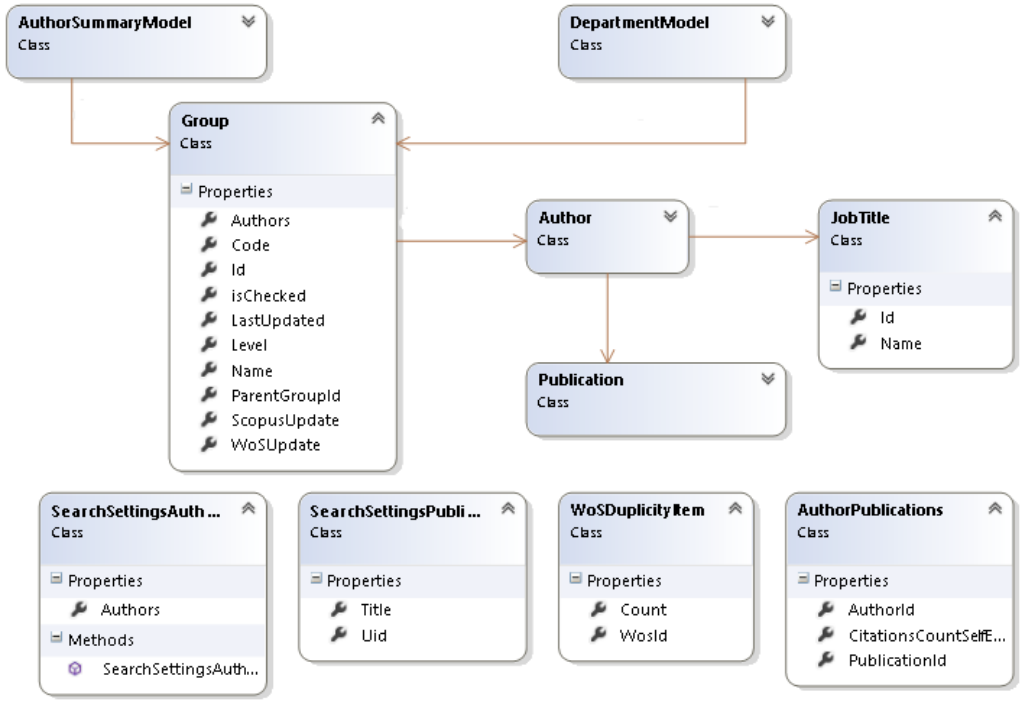
- **Prezentační vrstva**
  - Prezentace výsledků uživateli formou ASP.NET Webforms aplikace.
  - Interakce se zobrazenými daty (filtrování, řazení).
  - Správa IS v admin části.
- **Aplikační vrstva**

- Rozhraní přístupu pro datové zdroje.
- Objektově-relační mapování datových zdrojů (databázové tabulky) na doménové objekty.
- Spouštění databázových dotazů a procedur.
- DAO pro práci s daty v lokální databázi (skrze SQL[14]).

- **Datová vrstva**
  - Lokální SQL Server.
  - Zdrojové databáze.

4.3 Doménové objekty

Každý IS pracuje s daty a právě ta je nutné během práce uchovávat v paměti serveru v určitých strukturách. K tomuto účelu slouží doménové objekty, které odráží podobu (atributy) objektů nesoucích data. Projekt *Entities* obsahuje právě takové strukury, které slouží pro reprezentaci dat položek z databáze, operace nad nimi a rovněž pro jejich zobrazení ve formě modelů pro prezentační vrstvu. Takový projekt je referencován v ostatních částech IS, které potřebují pro práci s daty využívat právě objekty daného typu.

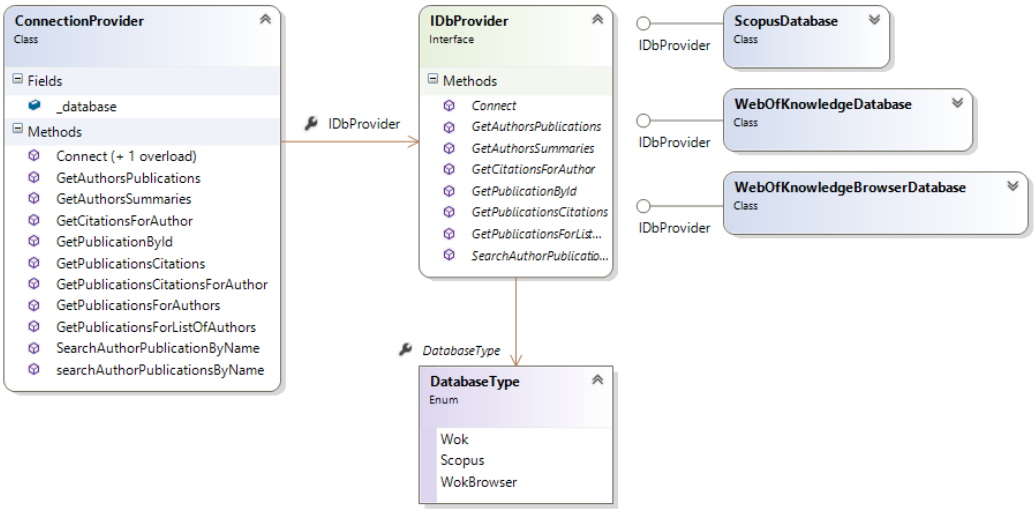


Obrázek 8: Doménové objekty a objekty nastavení

[14]Structured Query Language - dotazovací jazyk pro relační databáze

4.4 IDbProvider - interface zdrojových databází

V využívání obecného a znovupoužitelného přístupu ke zdrojovým databázím slouží třída *Common.ConnectionProvider*, která pracuje s objekty implementujícími rozhraní *IDbProvider*. Toto rozhraní definuje signatury metod pro získávání dat ze zdrojových databází. Třída *ConnectionProvider* je ústředním bodem komunikace s modulem (knihovnou tříd) APIObject.

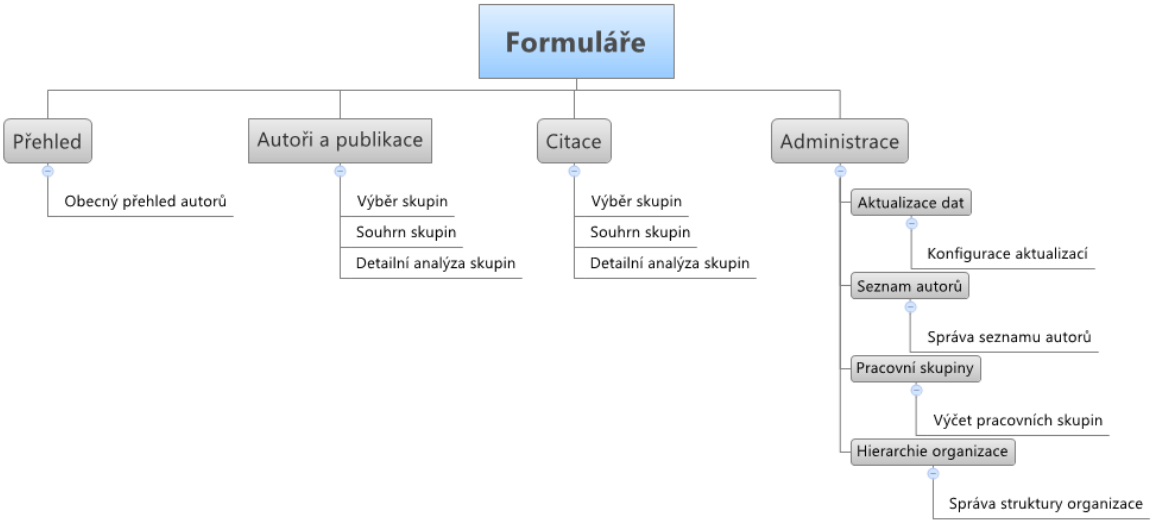


Obrázek 9: Třídní diagram použití rozhraní IDBProvider

4.5 Aplikace pro prezentaci dat

Pro účely prezentace dat se nabízí využití řešení ASP.NET Webforms, které poskytuje možnost vytvoření webové aplikace složené z připravených komponent s možností úprav pro potřeby projektu. Data lze prezentovat a pracovat s nimi pomocí předdefinovaných komponent GridView, TextBox, Button atd. Takto vytvořená prezentační vrstva umožňuje jednoduše a efektivně pracovat s událostmi na stránce, které mohou spouštět funkce doménové logiky aplikace včetně získávání dat z databáze. Pro získání příjemného vzhledu a ovládání může být pro klientskou část využita knihovna Bootstrap, která poskytuje CSS pro ovládací prvky a rovněž funkce pro jejich obsluhu. Tato knihovna je velmi rozšířená a oblíbená při práci s HTML, CSS a JavaScriptem.

Prezentace získaných dat má jednoduchý charakter formulářů s interaktivními tabulkami (komponenta GridView). Nesmí chybět nastavení zobrazení pro skupiny autorů a řazení zobrazených dat. Při navrhování systému byla brána v potaz možná nepřehlednost tabulek vzhledem k velkému počtu dat, kdy by se systém musel rozšířit o detailní pohledy na informace obsažené v tabulkách. Finální rozhodnutí však přináší pouze formuláře pro základní společný přehled obou



Obrázek 10: Rozložení formulářů (šedá barva) a popis obsahu (Zdroj: Xmind)

zdrojů (WoS, Scopus) a typů dat (autoři, citace). Další formuláře obsahují tabulky s detailnejší analýzou pro každou z diplomových prací.

Pro jednoduchou orientaci je do master page (jednotná "šablona"pro formuláře) začleněno navigační menu, kterým lze jednoduše přecházet mezi formuláři. Prvky stránky jsou responsivní, je tedy ošetřena optimalizace pro různá rozlišení či zobrazovací zařízení. Vývoj a debug nicméně probíhá v prohlížeči Chrome 56.0.

5 Implementace

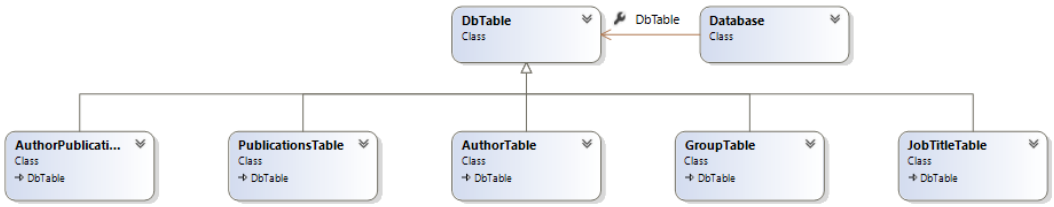
Tato kapitola popisuje implementaci informačního systému a strukturu řešení. Detailnější pohled na strukturu projektu lze vidět na obrázku 20 v příloze.

Popis jednotlivých projektů v řešení:

- **APIObjects:**
  - **Common:** společný projekt poskytující rozhraní pro práci s databázemi.
  - **Entities:** Doménové objekty.
  - **Scopus:** Implementace přístupu ke Scopus DB skrze API.
  - **WebOfKnowledge:** Implementace přístupu k WoS DB skrze API.
  - **WebOfKnowledgeBrowser:** Implementace nástroje získávajícího data z prohlížeče aplikace WoS.
- **DataAccess:** Implementace přístupu k interní DB
- **Web:** ASP.NET WebForms aplikace pro prezentaci výsledků

5.1 Data Access Objects

DAO s SQL příkazy (např. *PublicationsTable*) a procedurami pro každou z databázových tabulek v projektu *DataAccess* implementují rozhraní *IDbTable* a využívají třídu *Database*, která poskytu sadu metod pro komunikaci s SQL databází.



Obrázek 11: Třídy Data Access Object

```
public String SQL_SELECT = "SELECT * FROM Publications";
public String SQL_SELECT_ID = "SELECT * FROM Publications WHERE Id= @Id;"
public string SQL_SELECT_PUBLICATION_ID_BY_SCOPUSID = "SELECT TOP 1 Id F..."
public string SQL_SELECT_PUBLICATION_ID_BY_WOSID = "SELECT TOP 1 Id FROM..."
public string SQL_SELECT_PUBLICATION_LIST_BY_WOSID = "SELECT * FROM Publ..."
public string SQL_SELECT_PUBLICATION_ID_BY_NAME = "SELECT TOP 1 Id FROM ..."
```

```
public String SQL_INSERT = "INSERT INTO Publications VALUES (@Name, @Pub..."
public String SQL_DELETE_ID = "DELETE FROM Publications WHERE Id= @Id;"
public string SQL_DELETETEMP_ID = "DELETE FROM PublicationsTemp WHERE Id..."
public String SQL_UPDATE = "UPDATE Author SET Name = @Name, Published = ..."
public String SQL_SELECT_BY_AUTHOR = "SELECT P.Id, P.Name, P.Published, ..."
public String SQL_UPDATE_SCOPUS_CITATIONS_COUNTS = "UPDATE Publications ..."
public string SQL_UPDATE_WOS_CITATIONS_COUNTS = "UPDATE Publications SET..."
public String SQL_SELECT_BY_LEVEL = "SELECT * FROM Author a JOIN [Group]..."
public String SQL_COUNT_NON_PROCEEDINGS_BY_AUTHOR = "select COUNT(*) fro..."
public String SQL_COUNT_PROCEEDINGS_BY_AUTHOR = "select COUNT(*) from db..."
public String SQL_COUNT_NON_PROCEEDINGS_BY_AUTHOR_WOS = "select COUNT(*)..."
public String SQL_COUNT_PROCEEDINGS_BY_AUTHOR_WOS = "select COUNT(*) fro..."
```

Výpis 4: Ukázka definic SQL příkazů pro metody DAO PublicationsTable

5.2 Objektově relační mapování

Pro transformaci dat z databáze do objektů entit je vytvořeno ORM a to pomocí vzoru Table data gateway (TDG). Tento způsob mapování vytváří pro každou z tabulek v databázi "bránu", která komunikuje s databází a získaná data poskytuje v podobě konkrétních objektů entit. Každý TDG obsahuje jak CRUD metody pro práci s db, tak konkrétní dotazy na záznamy potřebné k prezentaci dat.

Tyto metody pro komunikaci s databází používají .NET třídu *SQLConnection*, pomocí které lze z definovaného connection stringu otevřít spojení s databází. K takto připojené databázi lze vytvářet a zasílat SQL příkazy. Získaná data jsou čtena pomocí třídy *SQLReader*, která podle typu získaných dat vytváří instance entit a plní je. Taková kolekce objektů je připravena pro využití v aplikaci. Tento přístup zajišťuje snadné úpravy v databázi nebo přidávání dalších tabulek, jelikož práve jedna TBG obstarává funkce pro práci s jednou tabulkou databáze a pořadí sloupců v tabulce databáze odpovída pořadí čtení hodnot SQLReaderem (výpis 5).

```
private Collection<Publication> Read(SqlDataReader reader)
{
      Collection<Publication> publications = new Collection<Publication>();

      while (reader.Read())
      {
            Publication publication = new Publication();
            publication.Id = reader.GetInt32(0);
            publication.PublicationName = reader.GetString(1);
            publication.Year = Convert.ToInt32(reader.GetString(2));
```

```
            publication.WokId = reader.IsDBNull(3) ? string.Empty : reader.
                GetString(3);
            publication.ScopusId = reader.IsDBNull(4) ? string.Empty : reader.
                GetString(4);
            publication.DocumentType = reader.GetString(5);
            publications.Add(publication);
        }
        return publications;
}
```

Výpis 5: Čtení hodnot získaných z databáze pomocí SQL Reader

5.3 Správa připojení k publikačním databázím

Třída *ConnectionProvider* poskytuje metody, které pomocí interface *IDbProvider* získávají daná data dle zvoleného zdroje (WoS, Scopus...), což umožňuje kdykoli jednoduše přidat další zdroje bez nutnosti změn implementace. Datový typ instance třídy implementující *IDbProvider* lze při připojování zvolit vstupním parametrem v metody *Connect*. V průběhu vývoje se tento přístup ukázal správným, jelikož bylo nutné v dalších fázích přidat právě nový zdroj, kterým je nástroj určený pro průchod webem Web of Science.

```
public class ConnectionProvider
   {
       public IDbProvider _database;
           public bool Connect(string dataBase){
                           switch(dataBase){
                                 case "WoS":
                                       _database = new WebOfKnowledgeDatabase();
                                       return _database.Connect();
                                 case "Scopus":
                                       _database = new ScopusDatabase();
                                       return true;
                                 case "WosB":
                                       _database = new
                                           WebOfKnowledgeBrowserDatabase();
                                       return true;}
                           return false;
           }
...
```

Výpis 6: Metoda třídy ConnectionProvider pro připojení k různým zdrojům

Způsob použití *IDbProvider* je možné vidět ve třídách *WebOfKnowledgeDatabase*, *WebOfKnowledgeBrowserDatabase* a *ScopusDatabase*, které obsahují implementace metody rozhraní. Ty jsou využívány pro získávání dat při aktualizacích. Takovou metodou je *GetAuthorsSummaries* na výpisu 7, která vrací objekty autorů s jejich publikacemi.

Tento zdroj informací však není schopný poskytnout některé důležité informace a to zejména citace, které jsou vyžadovány pro diplomovou práci Jiřího Littnera. Tento cíl je schopná splnit implementace průchodu webem WoS, jehož implementace je v projektu logicky zařazena jako třetí nezávislý zdroj dat viz 6.

```
public List<Author> GetAuthorsSummaries(List<Author> names)
{
    List<Author> summaries = _database.GetAuthorsSummaries(names);
    return summaries;
}
```

Výpis 7: Ukázka obecné implementace získání dat skrze ConnectionProvider

5.4 Web of Science

Prvotním návrhem získávání dat je implementace vlastního SOAP klienta, která je obsažena ve třídách WoSConnector a WoSSearcher projektu APIObjects.WebOfKnowledge. WoSConnector je zodpovědný za autorizaci a připojení k databázi Web of Science. Do vytvořeného HTTP požadavku je vložena autorizační SOAP zpráva a je odeslán na webovou autorizační službu. Odpověd serveru na takový požadavek obsahuje SID klíč pro další komunikaci. Požadavky je nutné zasílat ze sítě VŠB-TUO, jelikož samotná autorizace probíhá na základě IP požadavku. WoSSearcher následně toto SID získá předáním v konstruktoru při tvorbě instance. Web of Science API však není hlavním zdrojem dat pro tutu databázi, tím je průchod samotnám vyhledávačem.

5.5 Implementace průchodu webem WoS

Pro tento na API nezávislý zdroj dat z databází Web of Science je vytvořena třída WoSSearcherBrowser, která také implementuje metody rozhraní IDbProvider jako API zdroje WoS a Scopus. Projekt APIObjects je tedy schopem dodávat data ze tří zdrojů, byť jsou dva z nich zaměřeny na Web of Science. Na rozdíl od obou předchozích způsobů nelze využít ani SOAP client ani XML REST požadavky, ale je třeba programově navštívit databázi skrze zasílání požadavků na vyhledávač aplikace.

5.5.1 Komponenta System.Windows.Forms.WebBrowser

Testy implementace používající *WebBrowser* ukázaly, že tato komponenta má problémy s přetečením paměti při velkém počtu procházených stránek (několik hodin procházení). Je tedy nutné vytvořit vlastní implementaci komunikace založené na HTTP GET a POST požadavcích.

Odesláním GET požadavku (výpis 8) na URL vyhledávače aplikace Web of Science lze získat HTML dokument z odpovědi serveru. Takový dokument je možné procházet pomocí XPath a získat tak HTML prvky stránky. Prvky mají své atributy a do těch je možné zapsat hodnoty, jako například do textového pole identifikátor autora. Následně je potřeba zahájit vyhledávání výsledků odesláním formuláře POST požadavkem. Stejným způsobem lze otevřít odkaz prvního výsledku hledání, kde jsou při průchodu vnitřního HTML stránky publikace vypsány veškerá potřebná data jako: název, rok vydání, autoři či typ dokumentu. Takto lze pokračovat přechodem mezi výsledky až k poslednímu. Data jsou mezitím udržována v paměti aplikace v podobě doménových objektů pro zápis do databáze.

```
public HttpWebRequest CreateGetRequest(string requestUrl)
      {
          var request = (HttpWebRequest)WebRequest.Create(requestUrl);
          request.Method = "GET";
          request.CookieContainer = _cookieContainer;
          request.Accept = "text / html,application / xhtml + xml,application
              / xml; q = 0.9,image / webp,*/*;q=0.8";
          request.KeepAlive = true;
          request.Host = "apps.webofknowledge.com";
          request.UserAgent = "Mozilla / 5.0(Windows NT 10.0; Win64; x64)
              AppleWebKit / 537.36(KHTML, like Gecko) Chrome / 56.0.2924.87
              Safari / 537.36";
          request.Headers.Add("Origin", "https://apps.webofknowledge.com");
          request.Headers.Add("Accept-Encoding", "gzip, deflate, br");
          request.Headers.Add("Accept-Language", "cs - CZ,cs; q = 0.8,en; q =
              0.6");
          request.Headers.Add("Cache-Control", "no-cache");
          return request;
      }
```

Výpis 8: Požadavek GET pro získání HTML dokumentu z URL Web of Science

5.5.2 Požadavky GET a POST

Komunkace se serverem je zajištěna skrze HTTP požadavky v metodě *SendRequestToUrl*, která vytvoří GET či POST požadavek (pro získávání či posílání dat) a odešle jej na danou URL. Získaný obsah v odpovědi serveru je dekodován a navrácen v podobě HTML.

```
public string SendRequestToUrl(string requestUrl, string type) {
     if (type == "GET")
          request = CreateGetRequest(requestUrl);
```

```
        else if (type == "POST")
            request = CreatePostRequest(requestUrl);
        using (var response = (HttpWebResponse)request.GetResponse()) {
            using (var sr = new StreamReader(response.GetResponseStream())) {
                content = sr.ReadToEnd();
            }
        }
        string decodedHtml = HttpUtility.HtmlDecode(content);
        return decodedHtml;
}
```

Výpis 9: Metoda spravující GET/POST požadavky

5.5.3 Získání výsledků pro autora

Každý z autorů má různý počet publikací a je nutné je získat všechny, proto je naimplementován postup, který výsledky pro každého autora postupně prochází a posílá je do parsovacích metod. Jak již bylo naznačeno výše, přepínat mezi publikacemi autora lze přímo z detailu každé publikace (vpřed a vzad).

Metoda *Browse* (výpis 10) třídy *WoSSearcherBrowser* obstarává logiku získávání dat pro autora. Jednotlivé kroky průchodu jsou následující:

1. Získání SID klíče pro vyhledávání z app.webofknowledge.com
2. Přesměrování na vyhledávač s použitím session ID
3. Nalezení vyhledávacího pole a vložení hodnoty autora (obrázek 1)
4. Vyhledání roletky s možnostmi vyhledávání a volba parametru (obrázek 1)
5. Odeslání vyhledávacího formuláře
6. Získání URL prvního výsledku a přesměrování na detail (obrázek 4)
7. Parsování výsledku a uchování informací
8. Použití URL tlačítka k posunu na další detail výsledku
9. Opakování bodu 6 a 7 k poslednímu výsledku

```
    public Author Browse(Author author)
    {
        //Get base document
        _responseHtml = SendRequestToUrl(_baseUrl, "GET");
```

```
            //Get SID from response
            _sid = _parser.GetSID(_responseHtml);
            //Home page URL
            string url = string.Format("https://apps.webofknowledge.com/
                WOS_GeneralSearch.do?product=WOS&SID={0}", _sid);
            //Send search request
            _responseHtml = SendRequestToUrl(url, "POST");
            //Get first item URL
            string firstItemUrl = _parser.GetFirstItemUrl(_responseHtml);
            CheckZeroResults(firstItemUrl);
            url = _baseUrl + firstItemUrl;
            //Get first result document
            var resultItemHtml = SendRequestToUrl(url, "GET");
            _currentHtmlDoc = resultItemHtml;
            //Loop thru results and retrieve publications
            BrowseResults();
            return _author;
        }
```

Výpis 10: Metoda pro získávání dat pro autora

5.5.4 Parsování hodnot

Metoda *BrowseResults* zajišťuje průchod mezi jednotlivými detaily publikací a posílá jejich HTML dokument k parsování. Parsování probíhá ve třídě *Parser*, která obsahuje metody definující XPath daného prvku a také metodu *GetValueFromHtml* pro získání hodnot z dokumentu dle zvolené XPath. Definice XPath pro prvky mohou mít v rámci prvku DIV jinou pozici (index pole). Tato situace nastává, když mají detaily publikace rozdílný počet vyplněných informací. V případech kdy není známa konkrétní XPath, je nastavena XPath rodičovského prvku a ten je následně prohledán pro nalezení XPath požadovaného prvku (výpis 11).

```
internal string GetPublished(string htmlString)
{
      string result = string.Empty;
      string xPath = string.Empty;
      for (int i = 0; i < 6; i++ ){
            xPath = "//div[@class='block-record-info block-record-info-source']/
                p[@class='FR_field'][" + i + "]";
            result = GetValueFromHtml(htmlString, xPath, true);
            if (result.Contains("Published")){
```

```
                return result;
            }
        }
        return result = string.Empty;
}
```

Výpis 11: Definice XPath pro získání hodnoty data publikování z dokumentu

Se souřadnicí XPath je vybrán HTMLNode prvek dokumentu, který obsahuje požadovanou hodnotu. Hodnoty jsou udržovány v kolekci doménových objektů datového typu *Entities.Publication*. Tato kolekce je celkovým výsledkem operace průchodu webem a je opakována pro každého autora. Dokončením tohoto procesu je získána hrubá data o autorech a jejich publikacích.

Například informace zobrazené pro vydání publikace jsou u různých publikací v různých formátech:

- AUG-SEP 2015
- September 2015
- SEP 2015
- 2015

Pro účely IS je dostačující hodnout rok vydání a z výčtu typů formátu je využit vzor zápisu roku u každého z formátů a sice pozicí posledních 4 znaků. Metoda *GetYearFromString* (12) ze všech těchto kombinací zápisu získává rok vydání a to tak, aby jej bylo možné konvertovat do datového typu integer, vložit do objektů autorů a následně do databáze. Tento atribut je totiž využíván pro selekci z databáze a také pro řazení a filtrování v prezentační vrstvě, což v datovém typu string není možné.

```
private int GetYearFromString(string p)
{
      if (p != string.Empty)
            return Convert.ToInt32(p.Substring(p.Length - 4, 4));
      else
            return 0;
}
```

Výpis 12: Získání roku ze všech formátů zápisu

5.6 Scopus Searcher

V rámci získávání dat ze zdroje Scopus je vytvořena implementace využívající API databáze. Složením požadavků lze zaslat dotaz pro vyhledání publikací dle ID autora nebo ID publikace. Parametry tvorby požadavků je pořadí prvního výsledku, počet dotázaných výsledků od prvního a samotné filtrovací parametry (výpis 14).

```
requestUri = scopusSearchApi+"start="+start+"&count="+count+"&query=";
switch (idType) {
      case IdType.AuthorId:
              requestUri += "au-id(";
              break;
      case IdType.ScopusId:
              requestUri += "scopus-id(";
              break; }
```

Výpis 13: Tvorba URI s parametry vyhledávání

5.6.1 Parsování publikací z XML dokumentu

Parsování dat z XML dokumentu z odpovědi serveru probíhá ve třídě *Parser* za použití knihovny Linq.XDocument [20]. Tato knihovna nabízí rozšířené funkce třídy XMLDocument. Scopus používá různá XML schémata a syntaxe,ke kterým je nutné přiložit reference.

XDocument namespace:

- XNamespace atom = "http://www.w3.org/2005/Atom";
- XNamespace dc = "http://purl.org/dc/elements/1.1/";
- XNamespace prism = "http://prismstandard.org/namespaces/basic/2.0/";
- XNamespace ctom = "http://www.elsevier.com/xml/ctom/dtd";
- XNamespace opensearch = "http://a9.com/-/spec/opensearch/1.1/";

```
if (entry.Element(dc + "identifier") != null)
{
      var publication = new Publication
      {
            ScopusId = entry.Element(dc + "identifier").Value.Split(':').Count()
                > 1 ? entry.Element(dc + "identifier").Value.Split(':')[1].Trim
                () : entry.Element(dc + "identifier").Value.Split(':')[0].Trim()
                ,
```

```
            Title = entry.Element(dc + "title") != null ? entry.Element(dc + "
                title").Value : string.Empty,
            PublicationName = entry.Element(prism + "publicationName") != null ?
                 entry.Element(prism + "publicationName").Value : string.Empty,
            DocumentType = entry.Element(prism + "aggregationType") != null ?
                entry.Element(prism + "aggregationType").Value : string.Empty,
            CoverDate = entry.Element(prism + "coverDate") != null ? Convert.
                ToDateTime(entry.Element(prism + "coverDate").Value) : DateTime.
                MinValue,
            Authors = entry.Elements(atom + "author").Select(a => new Author
            {
                 ScopusId = a.Descendants(atom + "authid").First().Value
            }).ToList()
        };
        publications.Add(publication);
}
```

Výpis 14: Parsování výsledků z XML dokumentu

5.7 Aktualizace a úprava dat

UpdateController je třídou obsahující implementaci aktualizací a migrace dat. Samotná aktualizace lokálních dat probíhá v pracovní metodě *DoWork* knihovny *BackgroundWorker* (výpis 15), která ve vlastním vlákně spouští postupně kroky aktualizace. Průběh a konec operace je zaznamenán pomocí odebíraných událostí *ProgressChanged* či *RunWorkerCompleted*.

```
bw = new BackgroundWorker { WorkerReportsProgress = true,
    WorkerSupportsCancellation = true };
bw.DoWork += bw_DoWork;
bw.ProgressChanged += bw_ProgressChanged;
bw.RunWorkerCompleted += bw_RunWorkerCompleted;
bw.RunWorkerAsync();
```

Výpis 15: Odběr událostí třídy BackGroundWorker

Části aktualizace a jejich metody (*UpdateController*):

- Publikace WoS (*UpdateWoSPublications* - výpis 16).
- Publikace Scopus (*UpdateScopusPublications*).
- Citace WoS (*UpdateWoSCitations*).

- Citace Scopus (*UpdateScopusCitations*).
- Odstranění duplicit (*RemoveDuplicities*).
- Přepočty pro autory (*UpdateAuthorCounts*).
- Překopírování dat z dočasných tabulek(*UpdatePrimaryTables*).

Data jsou stahována po katedrách a ukládána do dočasných tabulek *PublicationTemp* a *AuthorPublicationsTemp*. V rámci zpracování kateder je měřen čas potřebný pro dokončení operace. Tyto časy jsou součástí kapitoly Měření, prezentace výsledků. Každá z částí aktualizací je vykonávána ve vlastní metodě, např. publikace ze zdroje Web of Science jsou získávány v metodě *UpdateWoSPublications* (výpis 16).

Části jsou prováděny postupně v daném pořadí a při výskytu chyby (pád internetového spojení) se celý proces opakuje, dokud není úspěšně dokončen. Postup jednotlivých částí je logován na straně serveru a to jak pro získání statistiky doby stahování, tak pro snadnou lokalizaci případných chyb.

```
 private void UpdateWoSPublications(List<Entities.Group> departments)
{
      foreach (var department in departments)
      {
            bw.ReportProgress(department.Id);
            var authors = _authorTable.SelectByGroup(department.Id).ToList();
            _provider.Connect("WosB");
            _sw.Start();
            var results = _provider.GetAuthorsSummaries(authors);
            _sw.Stop();
            //Insert results to temporary DB tables
            TemporaryInsert(results);
            _groupTable.UpdateBit(department.Id,"wos", true);
      }
}
```

Výpis 16: UpdateWoSPublications - metoda obstarávající aktualizace publikací WOS

Metoda *RemoveDuplicities* (výpis 17) pro odstranění případných duplicitních záznamů v tabulce *PublicationsTemp* je pátým krokem aktualizace dat. Vybrány jsou všechny duplicitní záznamy, vybráno jedno ID, které se přiřadí do všech zúčastněných relací a jsou smazány zbylé duplicitní záznamy.

```
private void RemoveDuplicities()
{
      var ids = _publicationsTable.SelectWoSDupIdsTemp();
      List<Publication> publications = new List<Publication>();
      foreach (WoSDuplicityItem id in ids)  {
            var results = _publicationsTable.SelectIds(id.WosId);
            var first = publicationResults.First();
            //Update authors new ID and delete duplicated id publications
            foreach (var publicationId in results) {
                  if (publicationId.Id == first.Id)
                        continue;
                  _authorPublications.UpdatePubTempId(publicationId.Id,
                     remainingId);
                  _publicationsTable.Delete(publicationId.Id);
            }
      }
}
```

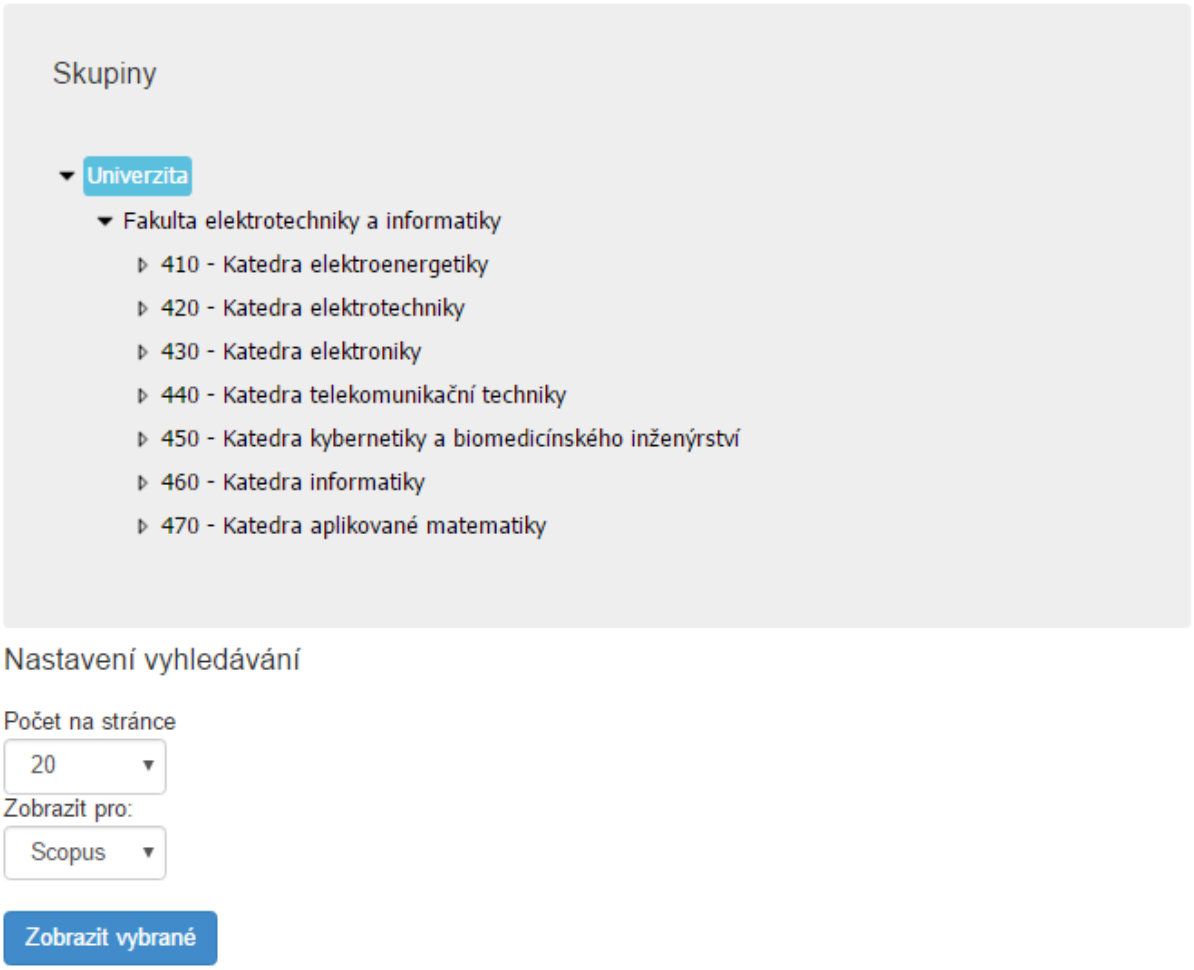
Výpis 17: RemoveDuplicities - metoda pro odstranění duplicit z dočasného úložiště

6 Formuláře

Kapitola obsahuje popis výsledných formulářů aplikace a jejich prvků.

6.1 Nastavení skupin

Jelikož je hlavním účelem systému prezentace informací o skupinách autorů, je zřejmé, že každý z formulářů musí obsahovat společný ovládací prvek pro volbu skupiny autorů (fakulta, katedra). K tomuto účelu slouží panel obsahující stromovou strukturu s možností volby takových skupin.



Obrázek 12: Navigační menu a ovládací panel pro volbu skupiny

Nastavení vyhledávání udává počet zobrazených výsledků na jednu stránku a pro který zdroj se data zobrazí (resp. pro oba). Ve společném zobrazení jsou v přehledech uváděny celkové hodnoty pro autory a skupiny.

6.2 Přehled

Společný přehled, který je zobrazen po příchodu do aplikace přináší základní pohled do statistik. Po zvolení katedry a fakulty je odeslán příslušný dotaz do databáze a získaná data jsou v podobě modelů zobrazena v tabulce seznamu autorů a jejich informací. Řazení lze měnit kliknutím na nadpis požadovaného sloupce, kde se zobrazí indikátor vzestupného či sestupného seřazení aktuálního sloupce (požití SortExpression). Pro vzhled ovládacích prvků a tabulky jsou použity CSS z knihovny Bootstrap.

Struktura tabulky přehledu:

- Pořadí - pořadí v rámci setřízení.
- Jméno - jméno autora.
- Login - login autora.
- Wos/Scopus: Čl. v časopisech: počet publikací v časopisech.
- Wos/Scopus: Čl. ve sbornících: počet publikací ve sbornících konferencí.
- Wos/Scopus: Počet citací bez autocitací - počet citací publikace.
- Wos/Scopus: H-Index - H-Index autora.

			Wos				Scopus			
Pořadí	Jméno	Login	Čl. v časopisech ▼	Čl. ve sbornících	Počet citací bez autocitací	H-Index	Čl. v časopisech	Čl. ve sbornících	Počet citací bez autocitací	H-Index
1	Kral Vladimir	abc0123	132	20	0	0	374	2	0	0
2	Slanina Zdenek	abc0123	92	15	0	0	4	14	0	0
3	Horak David	abc0123	70	15	0	0	32	37	0	0
4	Snasel Vaclav	abc0123	62	291	0	0	290	332	0	0
5	Cerny Martin	abc0123	58	33	0	0	16	48	0	0
6	Novak Tomas	abc0123	56	49	0	0	15	57	0	0
7	Kovar Petr	abc0123	35	6	0	0	22	0	0	0
8	Zelinka Ivan	abc0123	30	155	0	0	170	159	0	0
9	Seidl David	abc0123	28	9	0	0	12	2	0	0
10	Sikora Tadeusz	abc0123	25	20	0	0	5	21	0	0

Obrázek 13: Stránka se společným přehledem

6.3 Analýza autorů

Podrobnější analýza publikační aktivity autorů se nachází na stránce *Autoři a citace*, která také poskytuje možnosti nastavení skupiny autorů pro zobrazení výsledků. Implementace těchto funkcionalit se nachází ve stránce *AuthorsSummary.aspx* a příslušném kontroleru *AuthorsSummaryController.cs*. Dodatečné styly stránky jsou obsaženy v souboru *AuthorsSummary.css*.

Samotná stránka se skládá ze dvou částí reprezentovaných tabulkami:

1. Souhrn pro skupiny - statistika hodnot pro jednotlivé skupiny.
2. Seznam autorů - výpis hodnot pro zvolenou skupinu autorů.

První z tabulek zobrazuje statistiky pro vybrané skupiny autorů. První tří sloupce zobrazují pořadí, název skupiny a kód skupiny. Další sloupce obsahují jednotlivé hodnoty pro skupiny dle zdroje, typu dokumentu či časového období.

Přehled pro katedry (celkem)

Pořadí	Katedra	Kód ▲	ČČ*	ČS**	2017	2016	2015	2014	2013	2017 - 2013	Ostatní
1	Katedra elektroenergetiky	410	657	1283	6	77	207	124	213	627	1313
2	Katedra elektrotechniky	420	184	661	1	90	139	85	130	445	400
3	Katedra elektroniky	430	121	422	8	47	51	83	42	231	312
4	Katedra telekomunikační techniky	440	304	1123	6	231	246	167	179	829	598
5	Katedra kybernetiky a biomedicínského inženýrství	450	538	1624	5	282	251	211	252	1001	1161
6	Katedra informatiky	460	1141	4371	28	465	444	571	620	2128	3384
7	Katedra aplikované matematiky	470	715	478	15	82	135	117	88	437	756

* Počet článků v časopisech
** Počet článků ve sbornících

Obrázek 14: Statistiky autorů pro jednotlivé katedry

Sloupce tabulky souhrnu pro skupiny:

- Pořadí - pořadí v rámci setřízení.
- Název - název skupiny.
- Kód - kód skupiny.
- ČČ - počet publikací v časopisech.
- ČS - počet publikací ve sbornících konferencí.
- 2017, ... ,2013* - Výčet hodnot pro každý rok za posledních 5 let.

- 2017-2013 - Součet hodnot pro každý rok za posledních 5 let.
- Ostatní - Součet hodnot z ostatních let (starších 5 let).

Informace poskytnuté pro jednotlivé autory vybrané skupiny autorů jsou zobrazeny na obrázku 15. Celkový počet sloupců je 16 a jde převážně o numerické hodnoty reprezentující publikační činnost autorů. Ovládací prvky umožňují různé řazení seznamu dle parametrů nebo přesměrování na seznam publikací autora v publikačních databázích (WoS nebo Scopus). V rohu tabulky je obsažena informace o čísle aktuální stránky a v patičce lze přepínat mezi jednotlivými stránkami.

Seznam autorů (celkem)

* Link - kliknutím na tlačítko přejít proběhne přesměrování na seznam publikací daného autora v publikační databázi

Stránka číslo: 1

Informace o autorovi			Počet dle typu dokumentu		Dle let 2013 - 2017					Součty		Link*	
Pořadí	Jméno	Katedra	Čl. v časopisech ▲	Čl. ve sbornících	2017	2016	2015	2014	2013	2017 - 2013	Ostatní	WoS	Scopus
1	Abraham Padath Ajith	460	527	435	4	53	63	82	60	262	700	Přejít	Přejít
2	Kral Vladimir	410	506	22	3	15	11	18	40	87	441	Přejít	Přejít
3	Snasel Vaclav	460	352	623	9	87	100	117	99	412	563	Přejít	Přejít
4	Zelinka Ivan	460	200	314	5	66	50	80	100	301	213	Přejít	Přejít
5	Moravec Pavel	460	182	74	2	17	7	2	10	38	218	Přejít	Přejít
6	Krejcar Ondrej	450	143	201	2	54	50	19	22	147	197	Přejít	Přejít
7	Voznak Miroslav	440	123	159	6	71	58	39	41	215	67	Přejít	Přejít
8	Haslinger Jaroslav	470	107	4	2	6	4	11	6	29	82	Přejít	Přejít
9	Dostal Zdenek	470	103	21	0	3	8	8	6	25	99	Přejít	Přejít
10	Horak David	470	102	52	1	14	12	17	9	53	101	Přejít	Přejít
11	Slanina Zdenek	450	96	29	0	14	10	10	8	42	83	Přejít	Přejít
12	Platos Jan	460	89	182	0	15	15	40	39	109	162	Přejít	Přejít
13	Gajdos Petr	460	89	64	2	19	12	20	23	76	77	Přejít	Přejít
14	Penhaker Marek	450	88	123	0	32	24	40	21	117	94	Přejít	Přejít
15	Brandstetter Pavel	430	87	125	8	18	16	31	22	95	117	Přejít	Přejít
16	Froncek Dalibor	470	78	1	4	8	4	3	8	27	52	Přejít	Přejít
17	Kromer Pavel	460	77	158	1	25	26	22	23	97	138	Přejít	Přejít
18	Cerny Martin	450	74	81	0	15	31	19	21	86	69	Přejít	Přejít
19	Novak Tomas	420	71	106	0	19	20	9	26	74	103	Přejít	Přejít
20	Jancar Petr	460	65	16	0	2	5	7	5	19	62	Přejít	Přejít
21	Kalus Rene	470	64	0	0	2	7	5	5	19	45	Přejít	Přejít

Obrázek 15: Seznam autorů analýzy publikační činnosti

Sloupce tabulky analýzy publikační činnosti autorů:

- Informace o autorovi:
  - Pořadí - pořadí v rámci setřízení.

  - Jméno - jméno autora.
  - Kód - kód skupiny.
- Počet dke typu dokumentu:
  - ČČ - počet publikací v časopisech.
  - ČS - počet publikací ve sbornících konferencí.
- Dle let x-y:
  - 2017, ... ,2013* - Výčet hodnot pro každý rok za posledních 5 let.
- Součty:
  - 2017-2013 - Součet hodnot pro každý rok za posledních 5 let.
  - Ostatní - Součet hodnot z ostatních let (starších 5 let).
- Odkazy:
  - WoS - odkaz do systému WoS
  - Scopus - odkaz do systému Scopus

6.4 Administrativní část

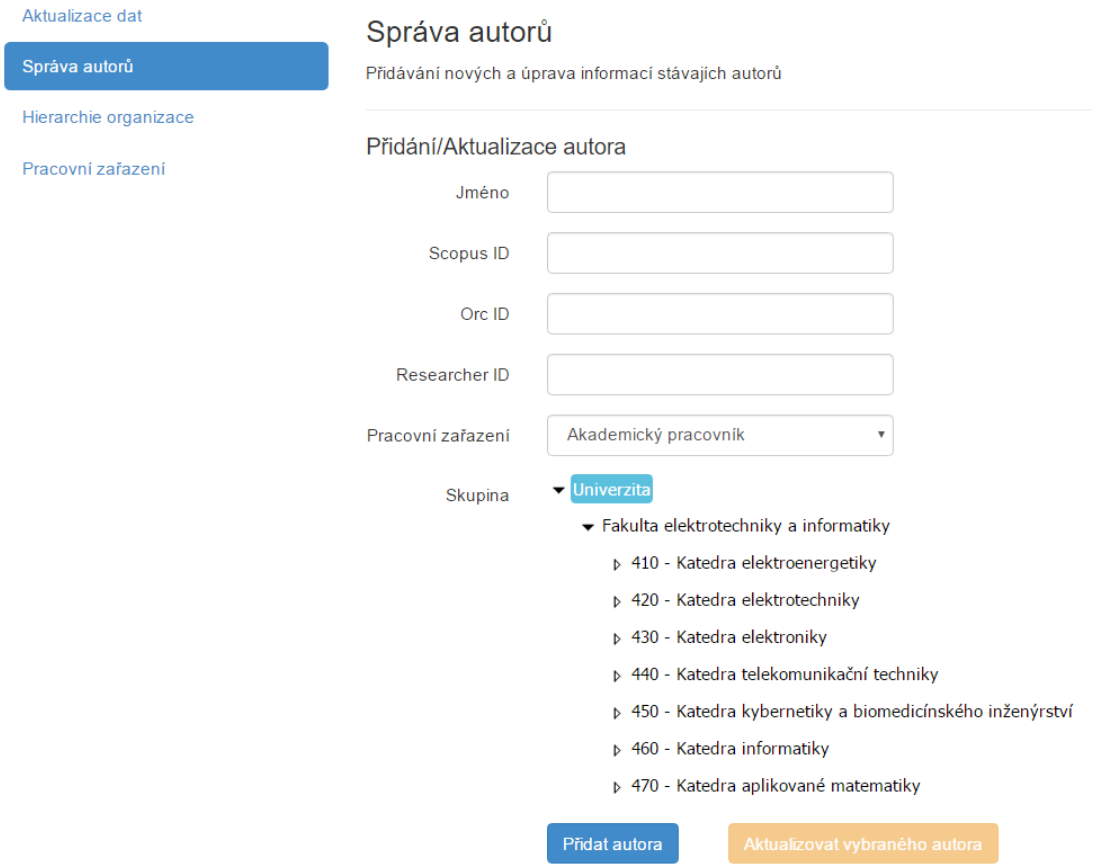
Administrativní část je složena ze čtyř sekcí.

1. **Aktualizace dat:**
   První sekce zobrazuje informace u stavu databáze a aktualizací.

2. **Správa autorů:**
   Sekce administrátora určená ke správě seznamu autorů a jejich informací. Lze zde přidávat další autory a jejich zařazení, či označit autora jako neaktuálního (změna organizace).

   Funkce:

   - Přidání/aktualizace autora.
   - Deaktivace autora.
   - Seznam autorů - upravitelný.



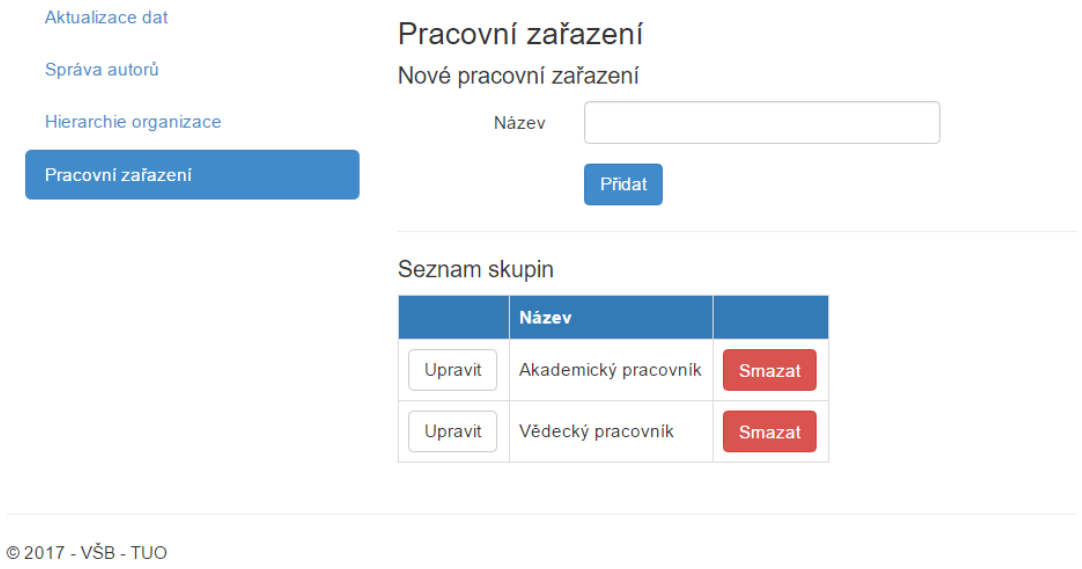
Obrázek 16: Správa autorů v administrativní části

3. **Hierarchie organizace [1]:**
   Pro úpravy stávající struktury organizace slouží sekce *Hierarchie organizace*. Je zde zobrazena stromová struktura aktuální hierarchie a funkce pro úpravu či rozšíření struktury.

   Funkce:

   - Přidání nové skupiny.
   - Aktualizace skupiny.
   - Smazání skupiny.
   - Seznam aktuálních skupin.

4. **Pracovní zařazení:**
   Poslední sekcí administrátora je správa pracovního zařazení pro autory. Momentálně existují zařazení: vědecký pracovník, akademický pracovník.



Obrázek 17: Správa pracovních skupin

7 Testování, měření a zhodnocení aplikace

Poslední kapitola popisuje testování a výsledná měření nad operacemi IS. Je zřejmé, že různé postupy a metody získávání dat mají rozdílné požadavky na časovou náročnost a také chybovost. Hlavní funkcionality systému prošly testováním rychlosti či několik hodin trvajících testů stahování dat. Tato kapitola dále nastíní metody a výsledky testování.

7.1 Odezva systému

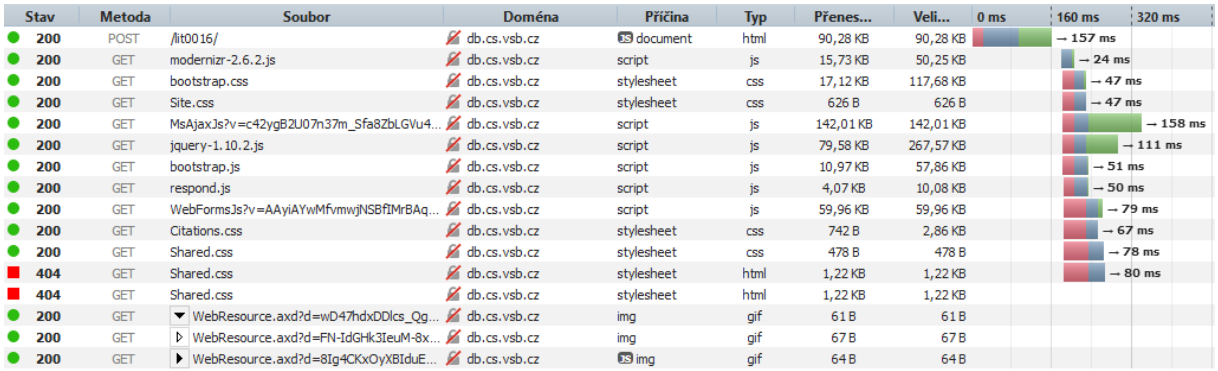
Rychlost odezvy serveru na uživatelský požadavek je důležitou a sledovanou vlastností systému, zejména při zobrazování velkého počtu dat. Jednotlivé kroky zpracování požadavku jsou pozorovány v nástroji Firebug[15] v přehledných tabulkách a grafech s možností detailního výpisu informací. Následuje popis testování rychlosti odezvy aplikace nasazené na školním serveru db.cs.vsb.cz. Toto testování probíhá ze zařízení mimo síť VŠB - TUO.

Získání formuláře *Přehled*:

Tento dokument o přibližné velikosti 90 KB je vytvořen a stažen v čase okolo 150 ms. Celková doba potřebná pro stažení celého obsahu stránky včetně skriptů se tak pohybuje pod hranicí jedné sekundy. Tento časový úsek je malý a uživatele nezdržuje.

Tabulka 2: Vlastnosti a naměřené hodnoty požadavku - formulář *Přehled*

Vlastnost	Naměřená hodnota
Počet záznamů autorů:	197
Velikost obsahu formuláře:	cca 90 KB
Vytvoření formuláře s daty:	70 - 80 ms
Stažení obsahu formuláře:	60 - 70 ms
Stažení a zobrazení celé stránky včetně skriptů:	800 ms - 1000 ms



Obrázek 18: Měření doby vyřízení požadavku na formulář s přehledem autorů

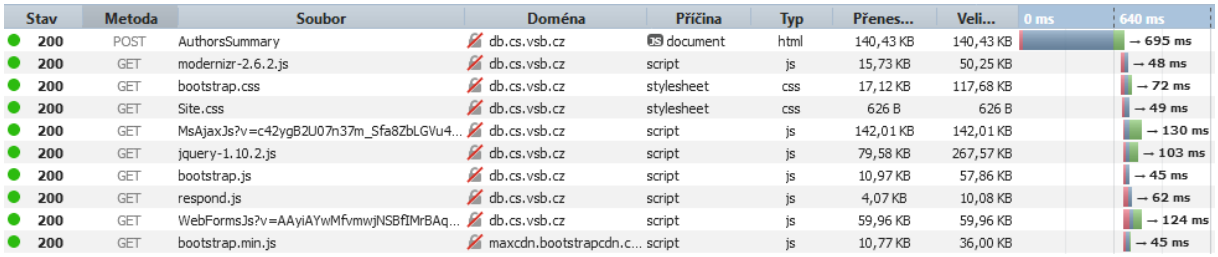
[15] Rozšíření prohlížeče Firefox

Získání formuláře *Autoři a publikace*:

Hlavní formulář s analýzou publikační činnosti má zhruba 140 KB a zpracování jeho dat a výpočtů trvá 750 - 800 ms. Celá stránka s analýzou a skripty je připravena za přibližně 1,5 sekundy.

Tabulka 3: Vlastnosti a naměřené hodnoty požadavku - formulář *Autoři a publikace*

Vlastnost	Naměřená hodnota
Počet záznamů autorů:	197
Velikost obsahu formuláře:	cca 140 KB
Vytvoření formuláře s daty:	750 - 800 ms
Stažení obsahu formuláře:	70 - 100 ms
Stažení a zobrazení celé stránky včetně skriptů:	cca 1500 ms



Stav	Metoda	Soubor	Doména	Příčina	Typ	Přenes...	Veli...	0 ms	640 ms
200	POST	AuthorsSummary	db.cs.vsb.cz	document	html	140,43 KB	140,43 KB		→ 695 ms
200	GET	modernizr-2.6.2.js	db.cs.vsb.cz	script	js	15,73 KB	50,25 KB		→ 48 ms
200	GET	bootstrap.css	db.cs.vsb.cz	stylesheet	css	17,12 KB	117,68 KB		→ 72 ms
200	GET	Site.css	db.cs.vsb.cz	stylesheet	css	626 B	626 B		→ 49 ms
200	GET	MsAjaxJs?v=c42ygB2U07n37m_Sfa8ZbLGVu4...	db.cs.vsb.cz	script	js	142,01 KB	142,01 KB		→ 130 ms
200	GET	jquery-1.10.2.js	db.cs.vsb.cz	script	js	79,58 KB	267,57 KB		→ 103 ms
200	GET	bootstrap.js	db.cs.vsb.cz	script	js	10,97 KB	57,86 KB		→ 45 ms
200	GET	respond.js	db.cs.vsb.cz	script	js	4,07 KB	10,08 KB		→ 62 ms
200	GET	WebFormsJs?v=AAyiAYwMfvmwjNSBfIMrBAq...	db.cs.vsb.cz	script	js	59,96 KB	59,96 KB		→ 124 ms
200	GET	bootstrap.min.js	maxcdn.bootstrapcdn.c...	script	js	10,77 KB	36,00 KB		→ 45 ms

Obrázek 19: Měření doby vyřízení požadavku na formulář s analýzou publikační činnosti autorů

7.2 Long-Term Testing

Vzhledem k časové náročnosti aktualizací dat je systém podroben několikahodinovým testům stahování všech dat. Tyto testy zejména poukazují na variabilitu zápisu hodnot v parsovaných dokumentech, špatně vyplněné nebo chybějící údaje v publikačních databázích Web of Science a Scopus. Testy obsahují logování chyb na straně serveru, podle kterého je možné snáz lokalizovat chyby. Každá taková chyba objevená při testování je následně ošetřena.

Tento typ testování rovněž odhalil problém s přetečením paměti při průchodu webem pomocí komponenty *WebBrowser*, která se pro daný typ úlohy ukázala jako nevhodná.

7.3 Měření doby stahování dat

Tabulka 2 zobrazuje přehled pro čas (zaokr. na minuty) potřebný ke stažení dat o publikacích pro všechny autory. První sloupec označuje kód katedry dané skupiny autorů, následuje počet autorů skupiny a počet stažených dokumentů pro Web of Science a Scopus.

Složený sloupec *Čas pro stažení dat*[16] obsahuje informace o časech pro různé zdroje (resp. přístupy) dat. První podsloupec zobrazuje hodnoty naměřené pro průchod webem Web of Science

[16] Čas měřen pomocí System.Diagnostics.Stopwatch

pomocí komponenty *WebBrowser*, druhý pro průchod pomocí HTTP GET/POST požadavků. Poslední sloupec udává čas potřebný pro získání dat z rozhraní Scopus.

Tabulka 4: Tabulka s testovanými daty a časy stažení

Skupiny	Počty			Čas pro stažení dat		
Kód kat.	Autoři ve sk.	Dok. WoS	Dok. Scopus	WoS Web	WoS Req.	Scopus API
410	21	815	1125	73 min	51 min	7min
420	19	397	448	15 min	6 min	3 min
430	11	176	367	10 min	4 min	2 min
440	24	569	858	31 min	23 min	6 min
450	41	894	1268	49 min	29 min	9 min
460	50	1580	3905	84 min	48 min	23 min
470	31	390	803	21 min	7 min	5 min
Celkem	**197**	**4821**	**8774**	**4 h 33 min**	**3 h 46 min**	**55 min**

Časy se pro opakované pokusy mění s odchylkou asi 10 %, a to kvůli nestálosti síťového provozu v takto dlouhých časových intervalech. Z tabulky výsledných časů lze pozorovat, že hlavním faktorem ovlivňujícím čas stažení je počet samotných publikací.

Velikostí skupiny autorů jsou časy ovlivněny minimálně, nicméně počty publikací jsou v tabulce zobrazeny bez duplicitních záznamů. Výskyt publikací se společnými autory v rámci početné skupiny tak může způsobit, že se prochází stejná publikace vícekrát, ale do databáze se po odstranění duplikací uloží pouze jednou. Z 500 získaných publikací tak může být použito například 250, což by ve výsledku zvýšilo průměrný čas získání publikace (tabulka 5) na dvojnásobek. Podle naměřených časů lze předpokládat, že tato situace nastala zejména u skupiny autorů katedry 470.

Výrazně nejrychlejší mezi WoS přístupy má Katedra elektrotechniky (420). Ostatní skupiny potřebovaly pro WoS zhruba stejný čas. Procházení jednotlivých detailů publikací v aplikaci

Tabulka 5: Tabulka průměrného času získání informací o publikaci

	Průměrný čas získání jedné publikace		
Kód katedry	WoS WebBrowser	WoS Req.	Scopus API
410	5,37 s	3,75 s	0,37 s
420	2,26 s	0,9 s	0,4 s
430	3,4 s	1,36 s	0,32 s
440	3,26 s	2,42 s	0,41 s
450	3,28 s	1,94 s	0,42 s
460	3,18 s	1,82 s	0,35 s
470	3,23 s	1,07 s	0,37 s
Průměr	**3,42 s**	**1,89 s**	**0,37 s**

WoS je tedy úzkým hrdlem celé operace. Přístup používající přímé požadavky je asi o 40 % rychlejší než využití komponenty *Winforms.WebBrowser*.

Časy pro získávání dat ze Scopus si jsou velice podobné a nevyskytují se zde výrazné odchylky od průměru. Doba potřebná pro získávání publikací ze Scopus téměř přímo úměrně roste s počtem procházených publikací bez vlivu na počet vzniklých duplicitních záznamů. Toto je důsledkem služby Scopus API, která do odpovědi na požadavek vkládá celý XML dokument se všemi výsledky pro autora a není tak nutné získávat jeden po druhém.

7.4 Časy algoritmů pro výpočty

- **Přepočet publikací autorů:**
  Funkce pro přepočet publikací autorů, která po stažení dat pro každého autora sestaví sumy počtů publikací dle typu a zdroje (WoS, Scopus) není sice spouštěna často, nicméně tvoří základ pro analýzu publikační činnosti autorů. Parametrem funkce je kolekce všech autorů (v tomto testu 197 autorů).

  Testování ukázalo, že tato funkce potřebuje pro dokončení průměrně 5 sekund, což je vzhledem k frekvenci volání více než přijatelné. Jednotlivá měření popisuje následující tabulka.

Tabulka 6: Rychlost výpočtů funkce pro statistiky autorů

Číslo měření	naměřený čas
1	4772 ms
2	5071 ms
3	5022 ms
4	4869 ms
5	4874 ms
6	4986 ms
7	5298 ms
8	5087 ms
9	5148 ms
10	4969 ms
Průměr	**5009 ms**

- **Výpočet hodnot pro sloupce s roky - autoři**
  Formulář s analýzou publikační činnosti obsahuje sloupce s počty publikací autora pro dané časové období. Tato funkce je volána při každém zobrazení dat ve formuláři *Autoři a citace*. Parametry jsou opět objekty autorů.

  Sloupce pro období:

  - Počet publikací pro každý rok za posledních 5 let
  - Suma počtu publikací za posledních 5 let
  - Suma počtu publikací starších 5 let

Následující tabulka zobrazuje naměřené hodnoty pro tuto funkci.

Tabulka 7: Rychlost výpočtů hodnot pro sloupce s počty publikací v daném období

Číslo měření	naměřený čas
1	23 ms
2	16 ms
3	18 ms
4	11 ms
5	14 ms
6	17 ms
7	11 ms
8	16 ms
9	15 ms
10	12 ms
Průměr	**15 ms**

- **Výpočet hodnot pro sloupce s roky - souhrn skupin**
  Stejné výpočty jsou prováděny v rámci zobrazení dat ve formuláři *Autoři a citace* pro tabulku souhrnu vybraných skupin.

  Sloupce pro období:

  – Počet publikací pro každý rok za posledních 5 let
  – Suma počtu publikací za posledních 5 let
  – Suma počtu publikací starších 5 let

Tabulka 8: Rychlost výpočtů hodnot pro sloupce souhrnu vybraných skupin v daném období

Číslo měření	naměřený čas
1	23 ms
2	16 ms
3	18 ms
4	11 ms
5	14 ms
6	17 ms
7	11 ms
8	16 ms
9	15 ms
10	12 ms
Průměr	**15 ms**

Naměřené hodnoty výpočtů sloupců dle období ukazují, že na rychlost vyřízení požadavků uživatele nemají výrazný vliv. Vzhledem k celkové době získávání a vykreslení stránky tak výpočty tvoří jednotky procent potřebného času.

8 Závěr

Požadavky kladené na informační systém této diplomové práce byly splněny. Největší část tvorby byla zaměřena především na analýzu publikačních databází a návrh cílového systému. Analýze, díky které bylo možné implementovat veškerou funkcionalitu systému, se věnují celkem dvě kapitoly práce. Požadavky kladené na rychlost, dostupnost a jiné důležité vlastnosti IS jsou realizovány za podpory kvalitního návrhu, školních hardwarových prostředků (aplikační server, databázový server, rychlost připojení k internetu) a vybraných technologií. Testováním systému byla odstraněna většina chyb a nedostatků.

Systém používá obecné principy implementace, proto je možné dále rozšiřovat funkcionalitu nebo přidávat další zdroje.

Vývoj IS Systém pro získávání informací o vědeckých publikacích skupin autorů mi přinesl nové znalosti z oblasti publikačních databází a technologií vybraných pro vývoj. Nemalým přínosem jsou zkušenosti při práci na projektu ve více lidech za použití nástrojů pro správu kódu. Komunikace s podporou databázových systémů či knihovnou byla rovněž přínosem pro obdobné situace v budoucnu.

Lze tedy konstatovat, že je tato diplomová práce přínosem jak pro autora, tak pro organizaci VŠB - TUO.

Tomáš Kucharczyk

Literatura

[1] Littner, Jiří, *Systém pro získávání informací o citačním ohlasu publikací pro skupiny autorů.* [DP]
Diplomová práce - VŠB-TUO FEI 2017

[2] *Kde hledat časopisy a konference pro publikování.* [online].
Veřejné informace o databázi Web of Science. Dostupné z: `http://veda.vse.cz/publikace/databaze-publikacni-cinnosti/kde-hledat-casopisy-a-konference-pro-publikovani/`

[3] *Elsevier.* [online]. [citace-3.1.2017].
Dostupné z: `https://www.elsevier.com/solutions/scopus`

[4] *Google Scholar.* [online].
Dostupné z: `https://scholar.google.com/intl/en/scholar/about.html`

[5] *Web of Science Web Services.* [online].
Dokumentace připojení k webovým službám Web of Science. Dostupné z: `http://science.thomsonreuters.com/tutorials/wsp_docs/soap/Guide/`

[6] *How to use HttpWebRequest and HttpWebResponse in .NET.* [online].
Dostupné z: `https://www.codeproject.com/Articles/6554/How-to-use-HttpWebRequest-and-HttpWebResponse-in-N`

[7] *System.Forms.WebBrowser.* [online].
Dokumentace k třídě WebBrowser. Dostupné z: `https://msdn.microsoft.com/en-us/library/system.windows.forms.webbrowser(v=vs.110).aspx`

[8] *Scopus API Docs.* [online].
Dokumentace přístupu k rozhraní Scopus. Dostupné z: `https://dev.elsevier.com/api_docs.html`

[9] *MSDN Academic Alliance.* [online].
Balík licencí od firmy Microsoft, který umožňuje zaměstnancům a studentům Katedry informatiky mít na školních i domacích počítačích nainstalovaný legální software, který je do produktu MSDN AA zahrnut. Dostupné z: `http://elms.cs.vsb.cz/`

[10] *Microsoft Visual Studio.* [online]
Dostupné z: `https://www.visualstudio.com/cs`

[11] *Extensible Markup Language (XML).* [online]
Dostupné z: `https://www.w3.org/XML/`

[12] *ASP.NET*. [online].
Dostupné z: https://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx

[13] Peter Drayton, Ben Albahari, Ted Neward, *C# v kostce - pohotová referenční příručka.* Překlad z anglického originálu "C# in a Nutshell"roku 2002

[14] *SQL Server Management Studio*. [online].
Dostupné z: https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms#sql-server-management-studio

[15] *SOAP Protocol*. [online].
Dostupné z: http://www.soapuser.com/basics1.html

[16] *SOAP UI*. [online].
Dostupné z: https://www.soapui.org/open-source.html

[17] *Google Chrome*. [online].
Dostupné z: https://www.google.com/chrome/browser/desktop/index.html

[18] Goossens, Michel, *The LaTeX companion,* New York: Addison, 1994.

[19] *Latex Table Generator*. [online].
Webový nástroj pro tvorbu Latex tabulek Dostupné z: http://www.tablesgenerator.com/

[20] *LINQ - XML*. [online]
XML parsing s pomocí Linq XDocument
Dostupné z: https://www.tutorialspoint.com/linq/linq_xml.htm

[21] *W3Schools*. [online]
Rozsáhlá internetová referenční příručka technologií vývoje
Dostupné z: https://www.w3schools.com/

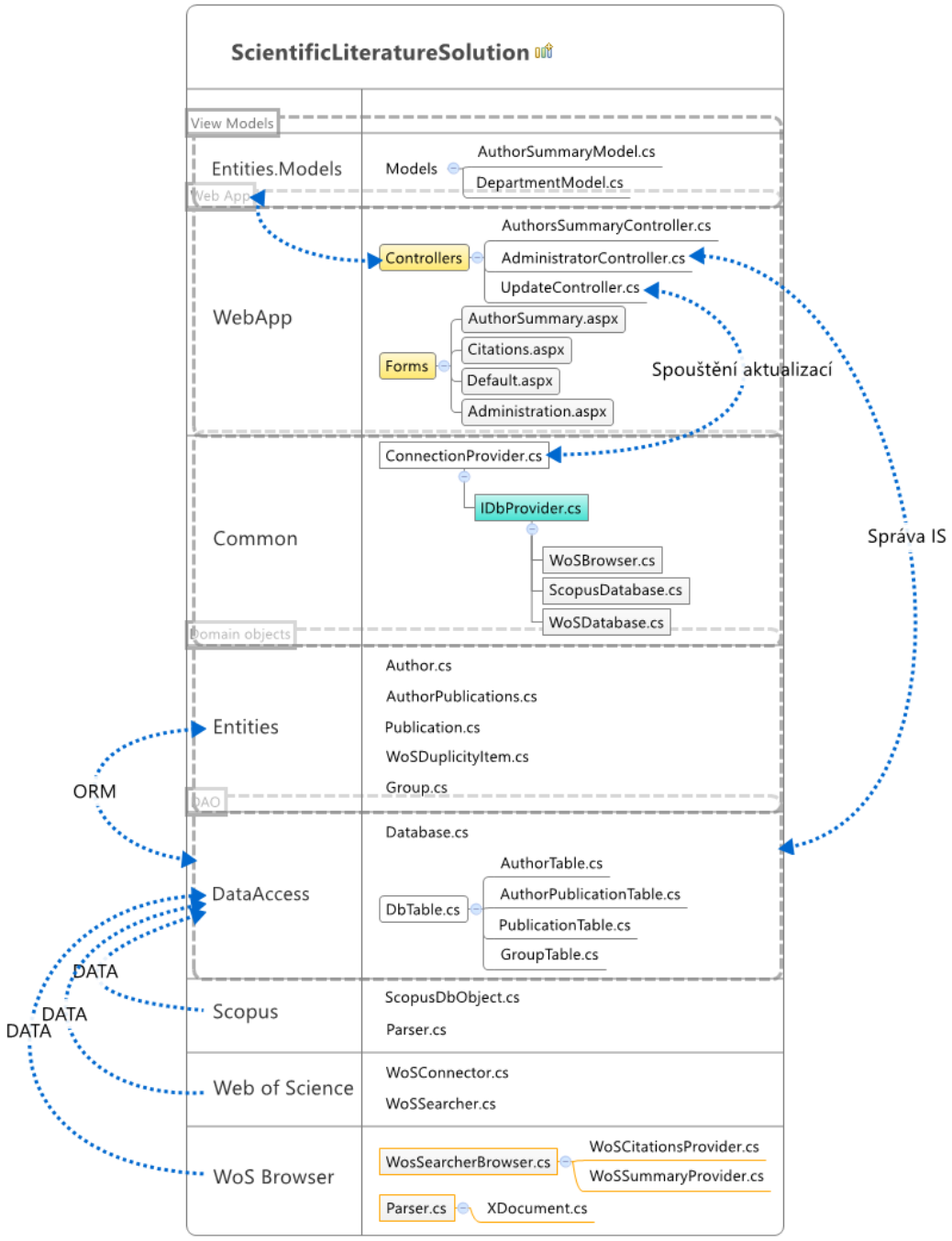
[22] *WoS User manual*. [online].
Uživatelský manuál pro použití vyhledávací aplikace webofknowledge Dostupné z: http://wokinfo.com/media/pdf/wos-corecoll_qrc_cz.pdf

[23] *Team Foundation Server*. [online].
Dostupné z: https://www.visualstudio.com/cs/tfs/?rr=https%3A%2F%2Fwww.google.cz%2F

[24] *Vlákna v C#*. [online]
Práce s vlákny v C# (threads, tasks, background workers).
Dostupné z: http://programujte.com/clanky/35-vlakna-v-c/

A Rozměrné přílohy

A.1 Diagram struktury projektu



Obrázek 20: Diagram struktury projektu - náhled na důležité části

B Obsah CD přílohy

- *ScientificLiteratureSolution* - složka s projekty.
  /Common
  /DataAccess
  /Entities
  /packages
  /ScientificLiteratureWeb
  /Scopus
  /WebOfKnowledge
  /WebOfKnowledgeBrowser

- *Scripts* - složka s SQL skripty.